# Adversarial Attacks in the Audio Domain

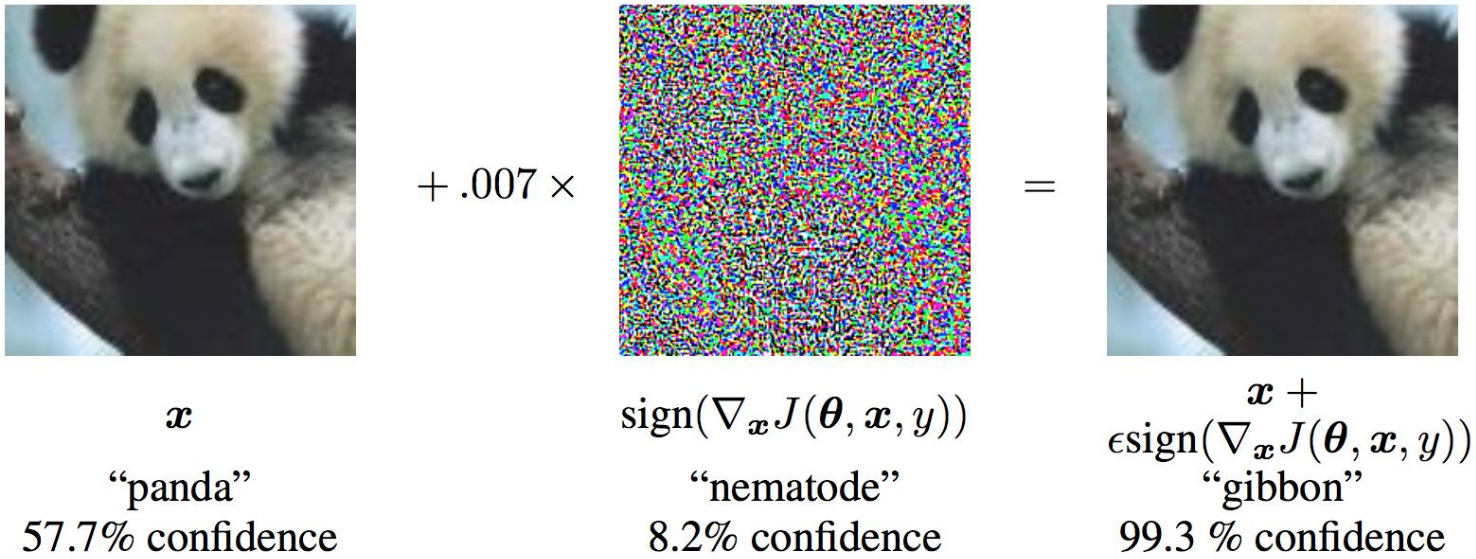
CS349 Machine Learning
Northwestern University
12.1.21

Patrick O’Reilly

github.com/oreillyp/adv_audio_intro

# Adversarial Examples Fool Neural Networks

# Adversarial Examples Fool Neural Networks



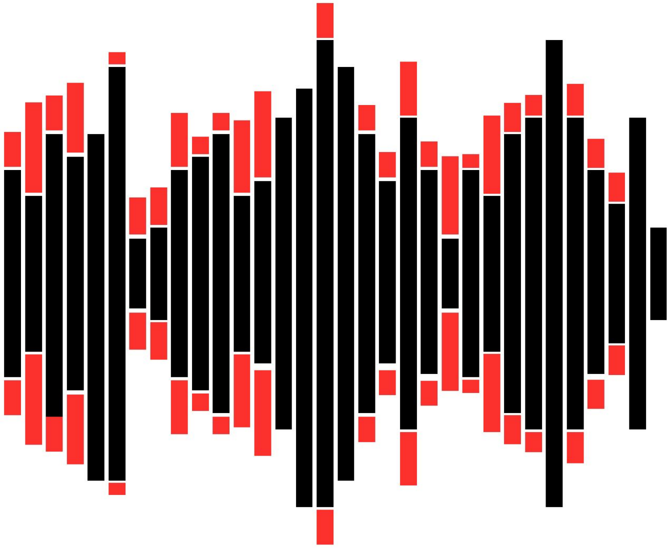

(Goodfellow et al. 2014)

# Adversarial Examples Fool Neural Networks

$x$

# Adversarial Examples Fool Neural Networks

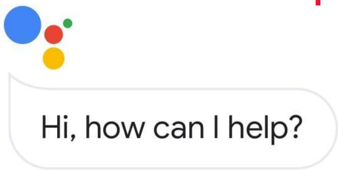
$$x + \delta$$



# Neural Networks Power Voice Interfaces

**Voice-based** machine-learning systems for authentication and control are common in products such as mobile devices, vehicles, and household appliances.




# What Systems Might Attackers Target?

# What Systems Might Attackers Target?

**Recognize** "Hey Alexa," "OK Google," "stop," "go," ...

***Wake-word detection, speech command recognition***

**Verify** a speaker's identity (against enrolled profile)

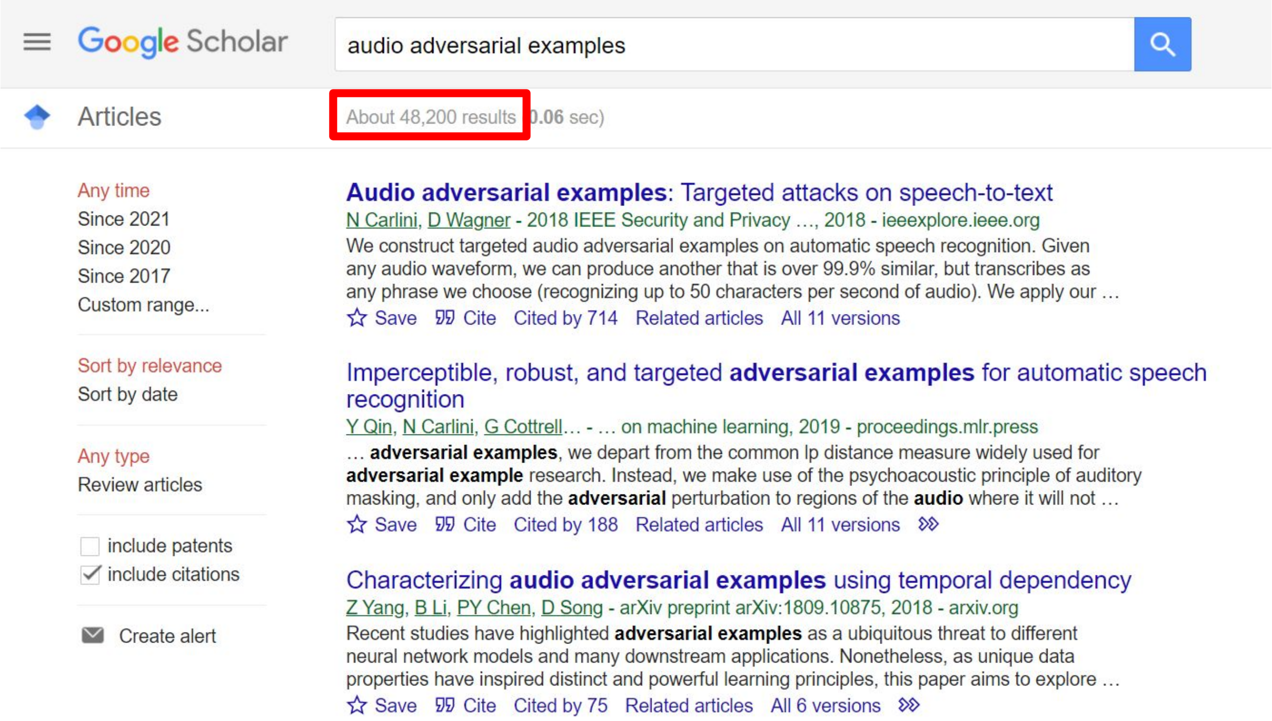
***Automatic speaker verification, speaker recognition***

**Transcribe** all incoming speech

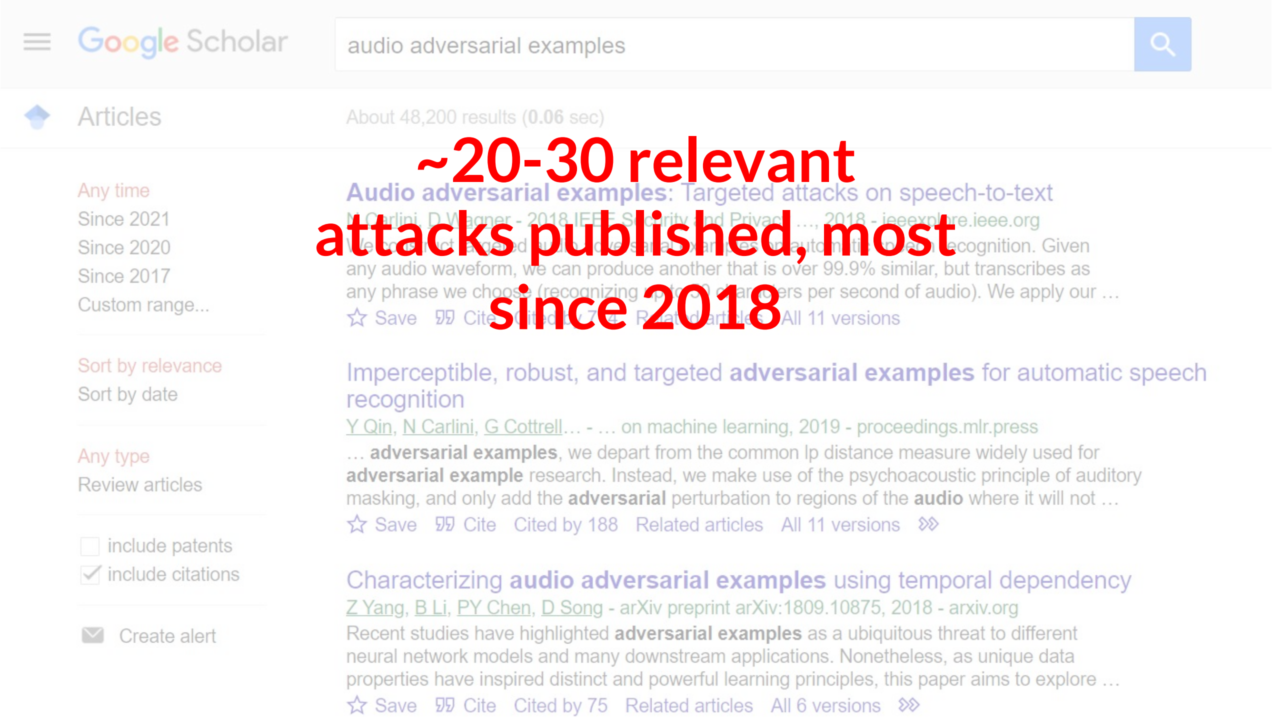
***Automatic speech recognition***

# ...Has Anyone Looked Into This?

# ...Has Anyone Looked Into This?



# …Has Anyone Looked Into This?




~20-30 relevant attacks published, most since 2018

# ...Has Anyone Looked Into This?


~20-30 relevant attacks published, most since 2018

A similar number of defenses have been proposed

# What Systems Might Attackers Target?

**Recognize** “Hey Alexa,” “OK Google,” “stop,” “go,” ...

*Wake-word detection, speech command recognition*

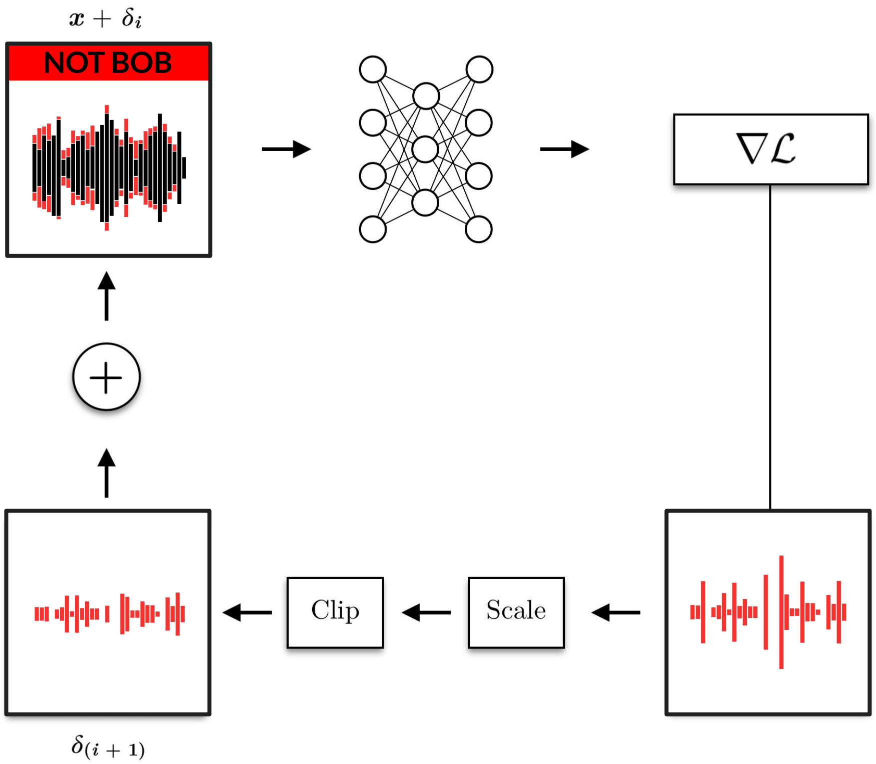
**Verify** a speaker’s identity (against enrolled profile)

*Automatic speaker verification, speaker recognition*

**Transcribe** all incoming speech

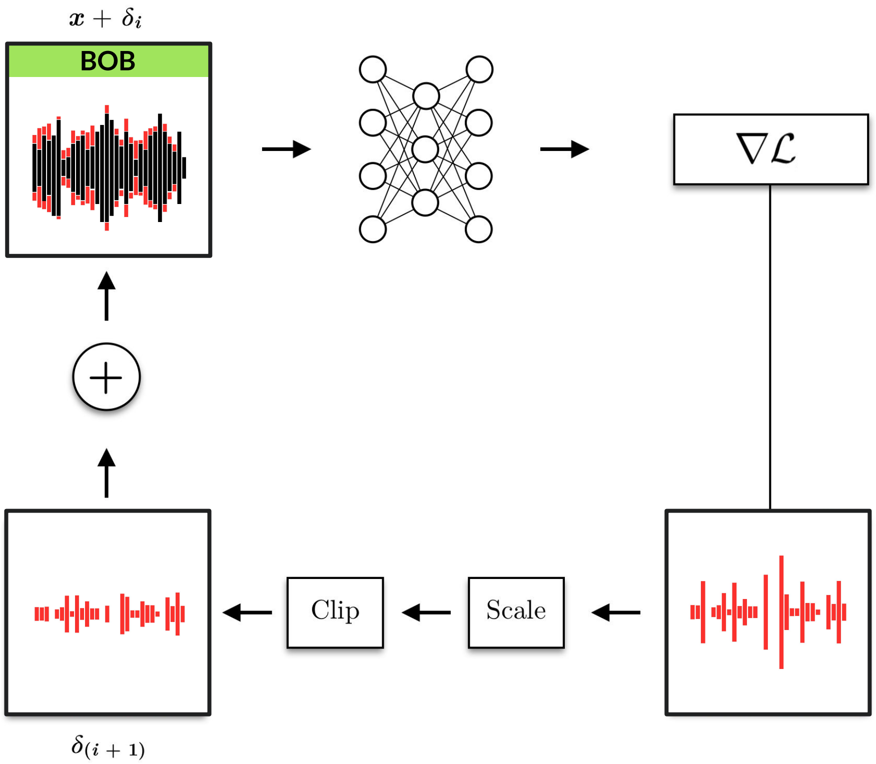
*Automatic speech recognition*

# How Do We Make Adversarial Examples?

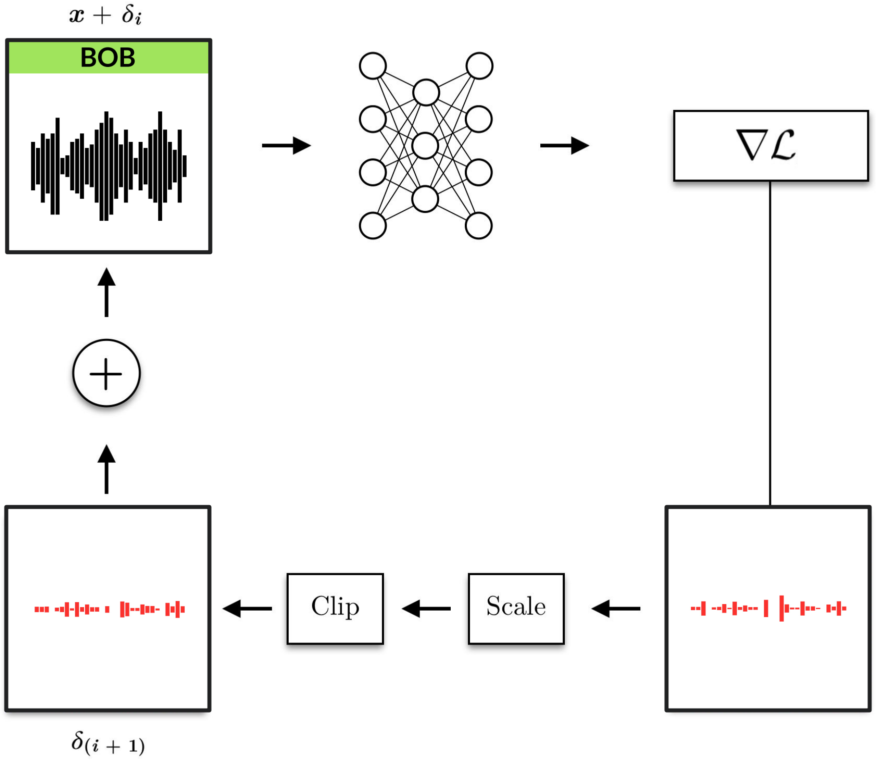


# How Do We Make Adversarial Examples?

$f(x + \delta_{final}) = $ **BOB**

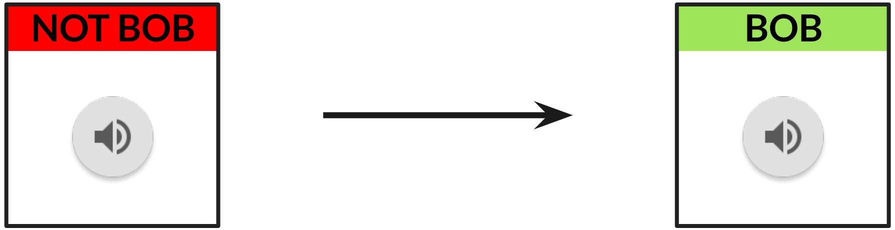


# How Do We Make Adversarial Examples?




$f(x + \delta_{final})$ = **BOB**

# How Should We Attack?



# Effective and Inconspicuous Over-the-Air Adversarial Examples with Adaptive Filtering

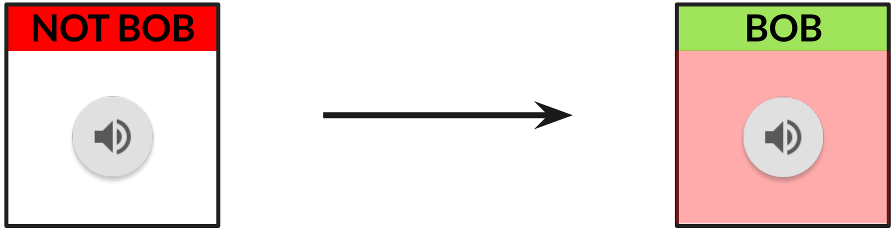
Patrick O'Reilly[1], Pranjal Awasthi[2], Aravindan Vijayaraghavan[1], Bryan Pardo[1]

*Submitted to ICASSP '22*

1. Northwestern University
2. Google Research

**interactiveaudiolab.github.io/project/audio-adversarial-examples.html**

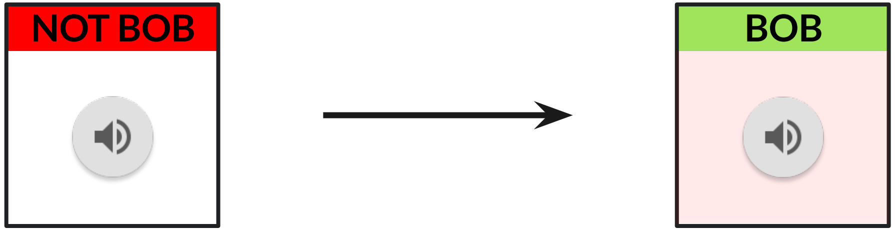
# How Should We Attack?




| | "Generic" |
|---|---|
| **Approach** | image-domain (sample-wise additive noise) |
| **Perceptual Regularization** | simple ($L_2$ penalty) |
| **Perceptual Quality** | poor (perturbation is obvious) |

# How Should We Attack?

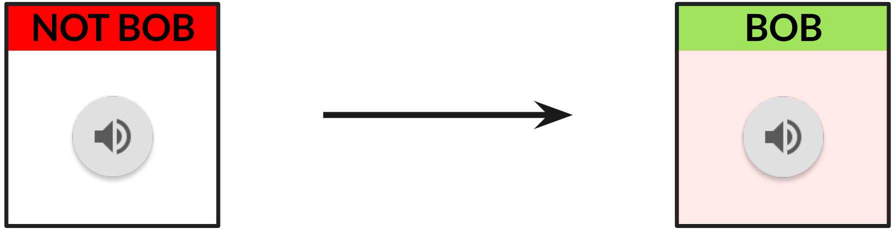
*Qin et al. (2019),
Szurley & Kolter (2019),
Dörr et al. (2020),
Wang et al. (2020)




| | "Generic" | Qin et al.* |
|---|---|---|
| **Approach** | image-domain (sample-wise additive noise) | image-domain (sample-wise additive noise) |
| **Perceptual Regularization** | simple ($L_2$ penalty) | complex (frequency masking loss) |
| **Perceptual Quality** | poor (perturbation is obvious) | good (perturbation is subtle) |

# How Should We Attack?

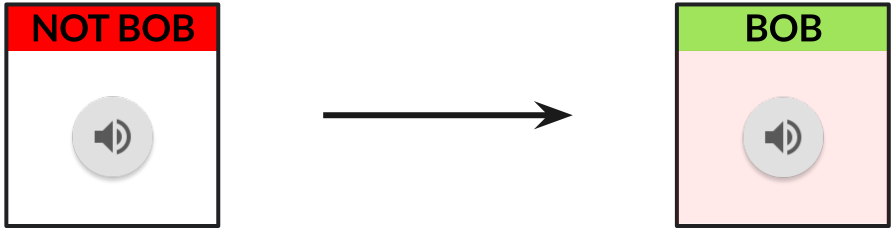
*Qin et al. (2019),
Szurley & Kolter (2019),
Dörr et al. (2020),
Wang et al. (2020)




| | “Generic” | Qin et al.* | Proposed |
|---|---|---|---|
| **Approach** | image-domain (sample-wise additive noise) | image-domain (sample-wise additive noise) | audio-domain (adaptive filtering) |
| **Perceptual Regularization** | simple ($L_2$ penalty) | complex (frequency masking loss) | simple ($L_2$ penalty) |
| **Perceptual Quality** | poor (perturbation is obvious) | good (perturbation is subtle) | good (perturbation is subtle) |

# How Should We Attack?

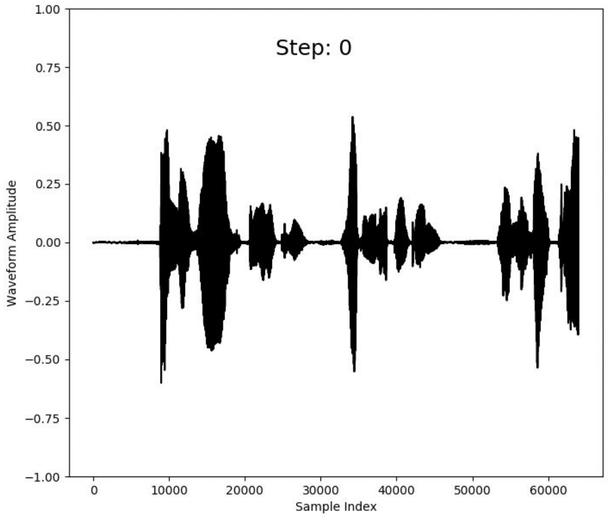
*Qin et al. (2019),
Szurley & Kolter (2019),
Dörr et al. (2020),
Wang et al. (2020)



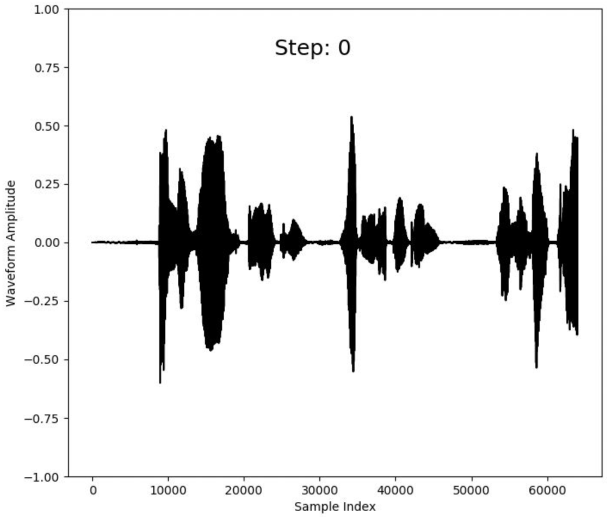

| | "Generic" | Qin et al.* | Proposed |
|---|---|---|---|
| **Approach** | ✗ | ✗ | ✓ |
| **Perceptual Regularization** | ✓ | ✗ | ✓ |
| **Perceptual Quality** | ✗ | ✓ | ✓ |

# How Should We Attack?

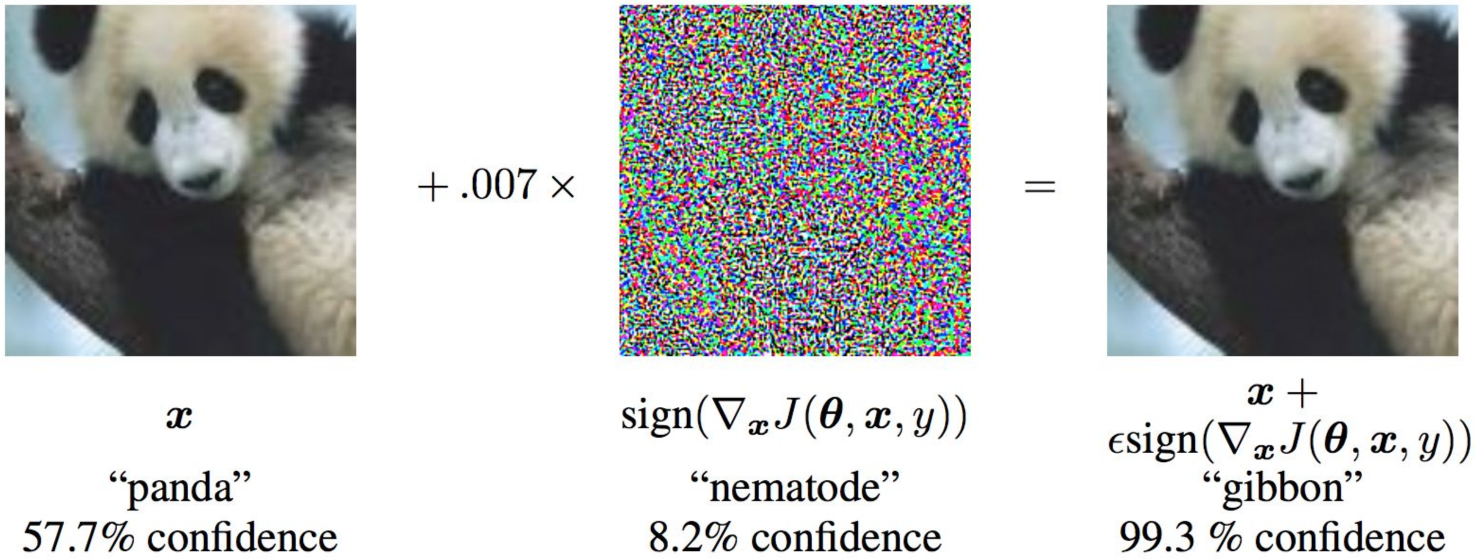
**Qin et al.***




**Proposed**



# Adversarial Examples Fool Neural Networks




(Goodfellow et al. 2014)

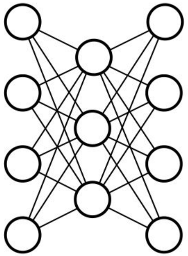
# Let’s Attack a Voice Interface

# Let’s Attack a Voice Interface: Pick a Task

**Speaker Verification:** confirm a speaker’s claimed identity (against enrolled profile)

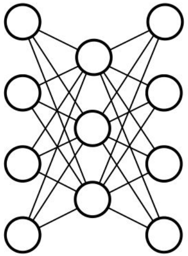
# Let’s Attack a Voice Interface: Pick a Task

We want a **large** and **accurate** model, as in many applications (e.g. mobile banking) speaker verification models are deployed in the cloud rather than on-device.

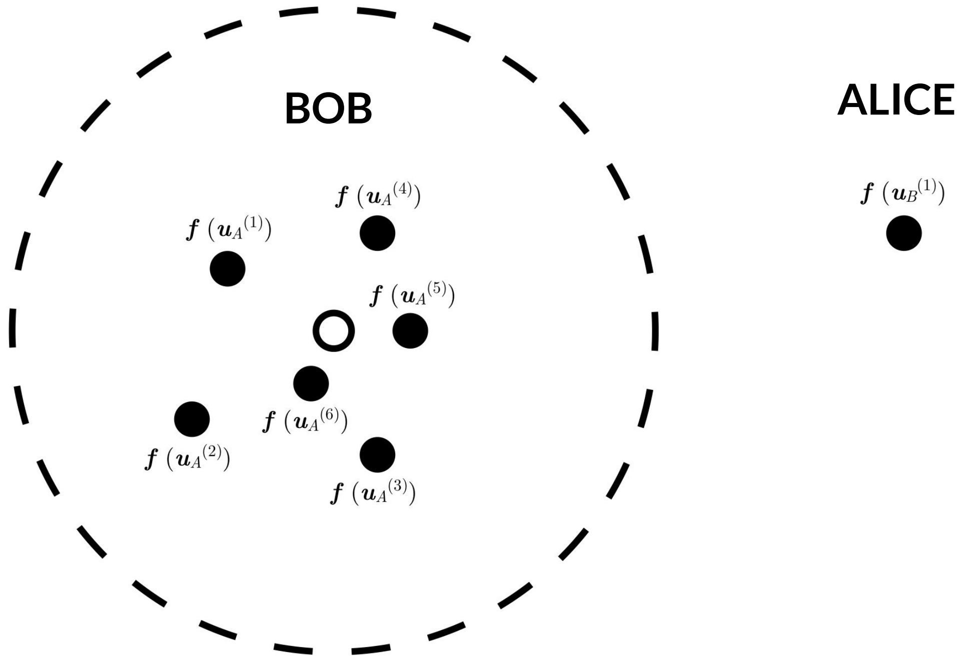


# Let’s Attack a Voice Interface: Pick a Task

Specifically, we’ll use the **ResNetSE34V2** model proposed by Heo et al. (2020), available at https://github.com/clovaai/voxceleb_trainer

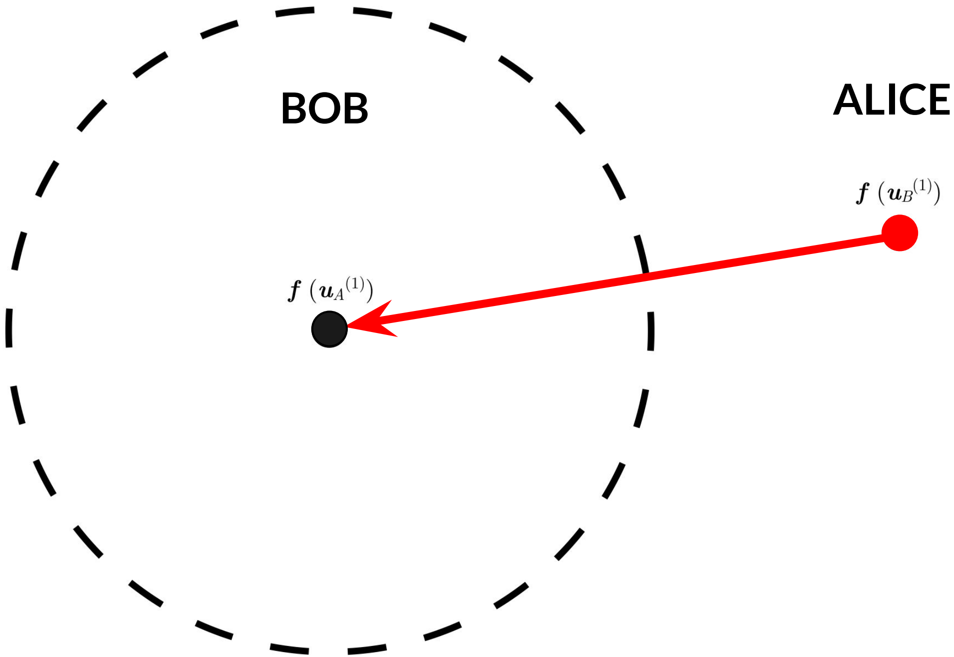


# Let’s Attack a Voice Interface: Pick an Objective



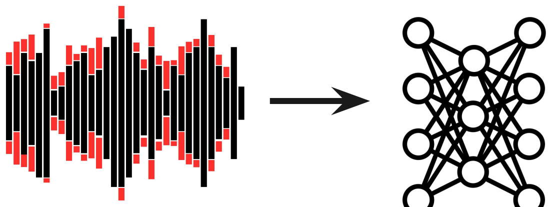
# Let's Attack a Voice Interface: Pick an Objective

Following Zhang et al. (2021), for the sake of simplicity we will attempt to spoof the embedding of a single utterance.



# Let’s Attack a Voice Interface: Pick a Setting

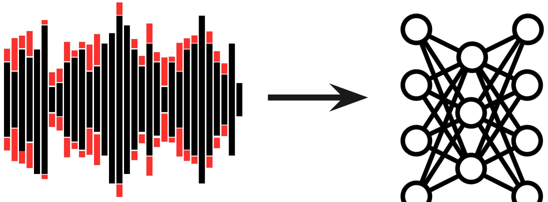
**Over-the-line setting**: the attack audio can be fed directly to the victim model over a purely digital channel.

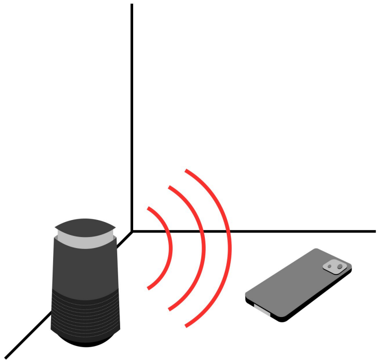


# Let's Attack a Voice Interface: Pick a Setting

**Over-the-line setting**: the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting**:  malicious audio is played through a speaker and received by a microphone before entering the victim model.

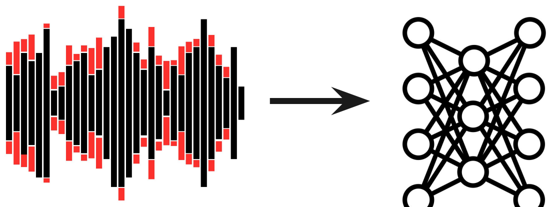


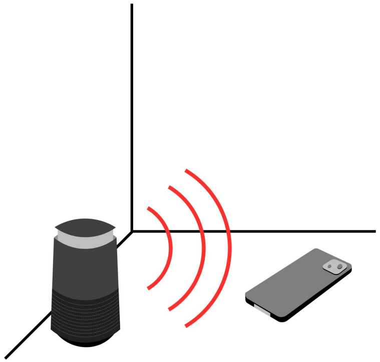


# Let's Attack a Voice Interface: Pick a Setting

**Over-the-line setting**: the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting**:  malicious audio is played through a speaker and received by a microphone before entering the victim model.





# Let's Attack a Voice Interface: Pick a Setting

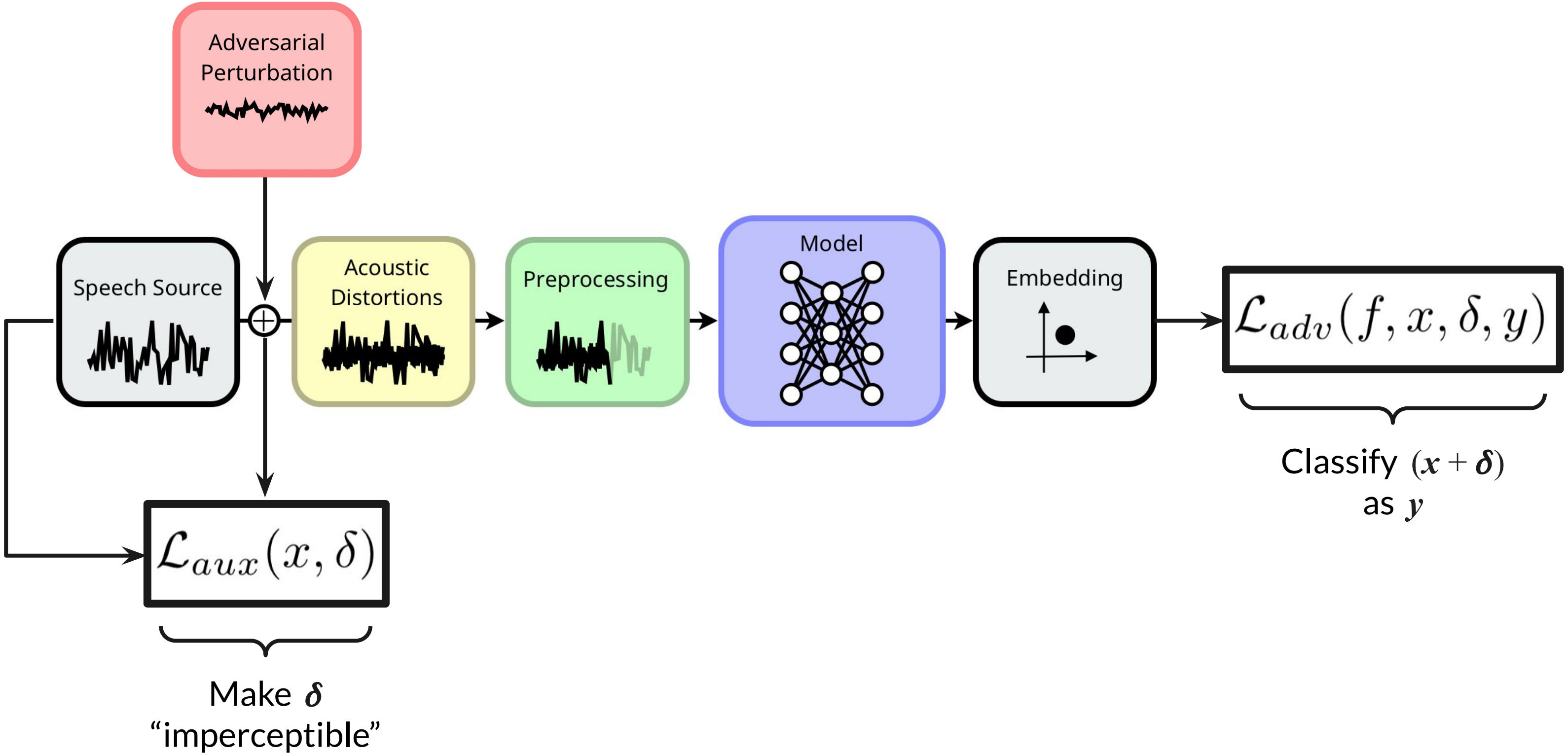
**Over-the-line setting**: the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting**:  malicious audio is played through a speaker and received by a microphone before entering the victim model.

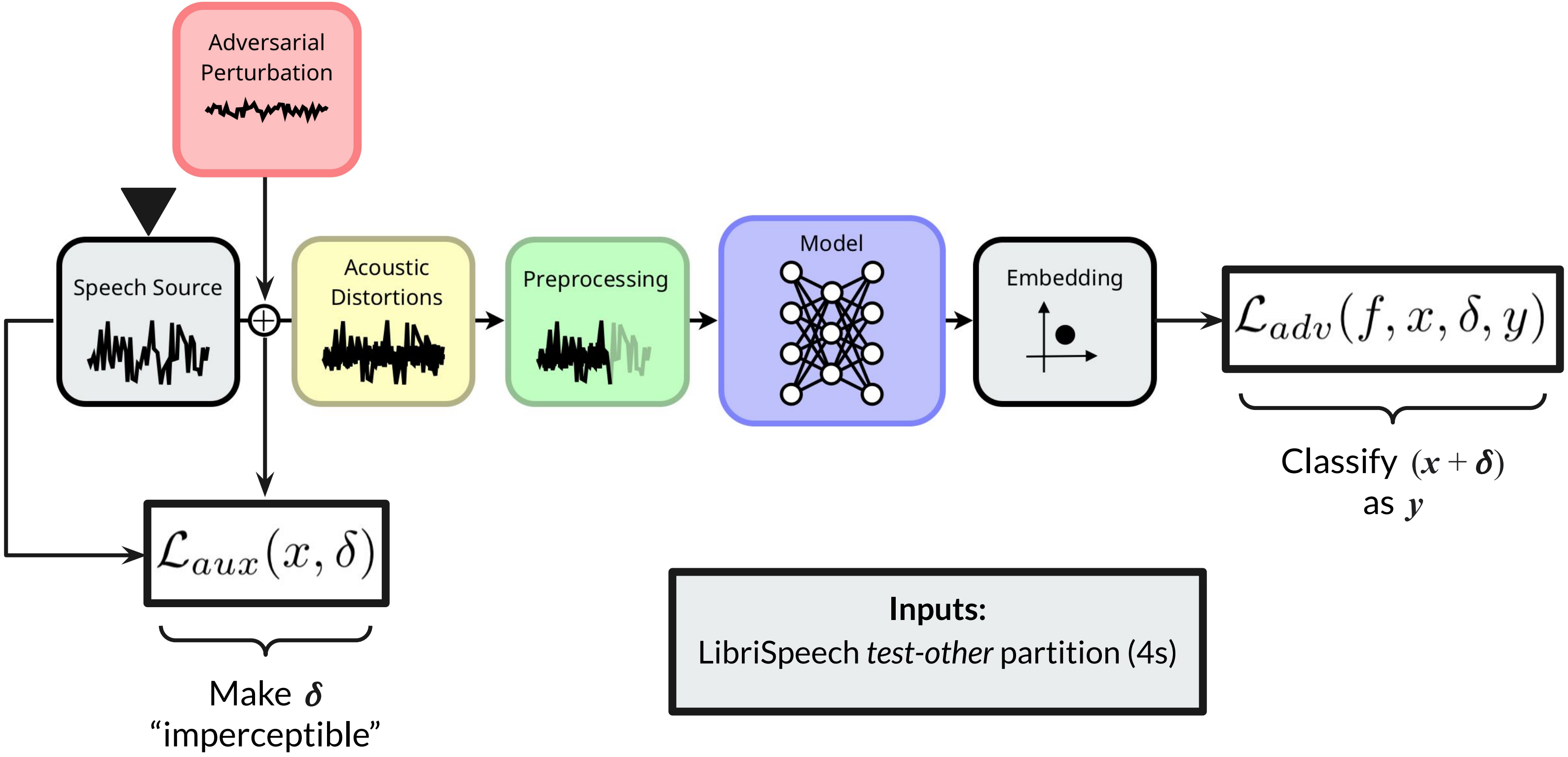
*Over-the-line*

*Over-the-air*

# Let's Attack a Voice Interface: System Design

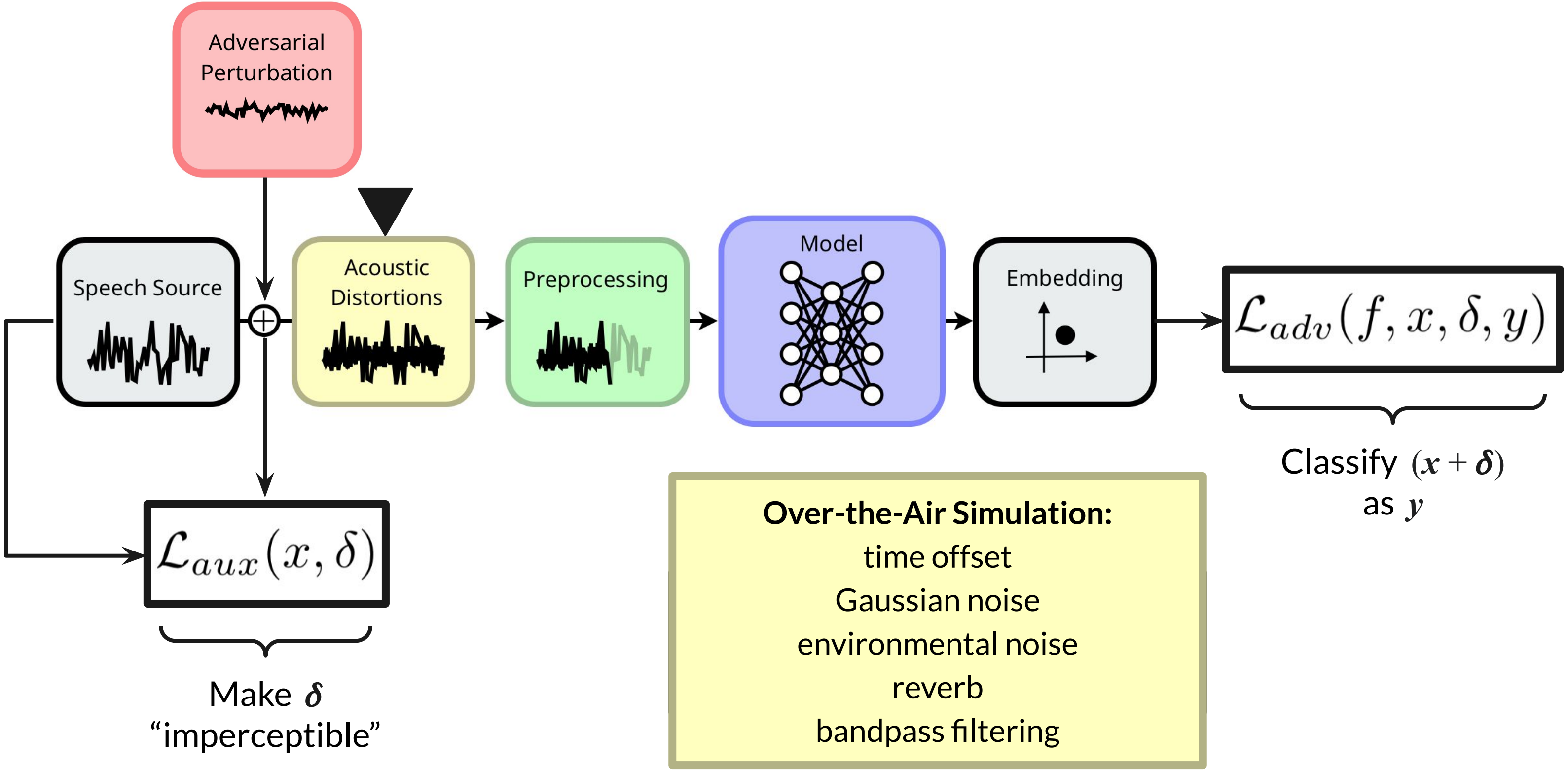


# Let's Attack a Voice Interface: System Design

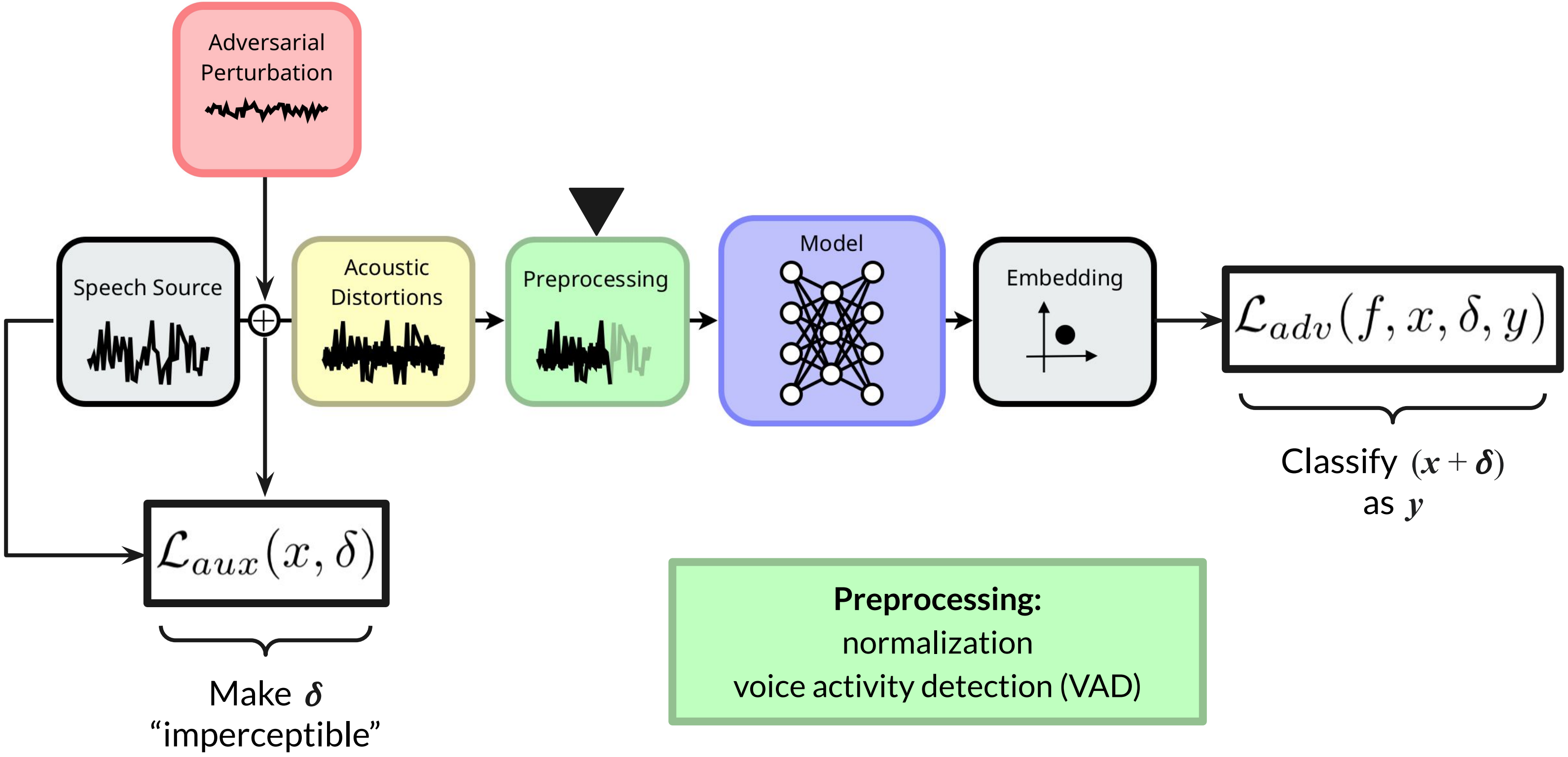



**Inputs:**
LibriSpeech *test-other* partition (4s)

# Let’s Attack a Voice Interface: System Design

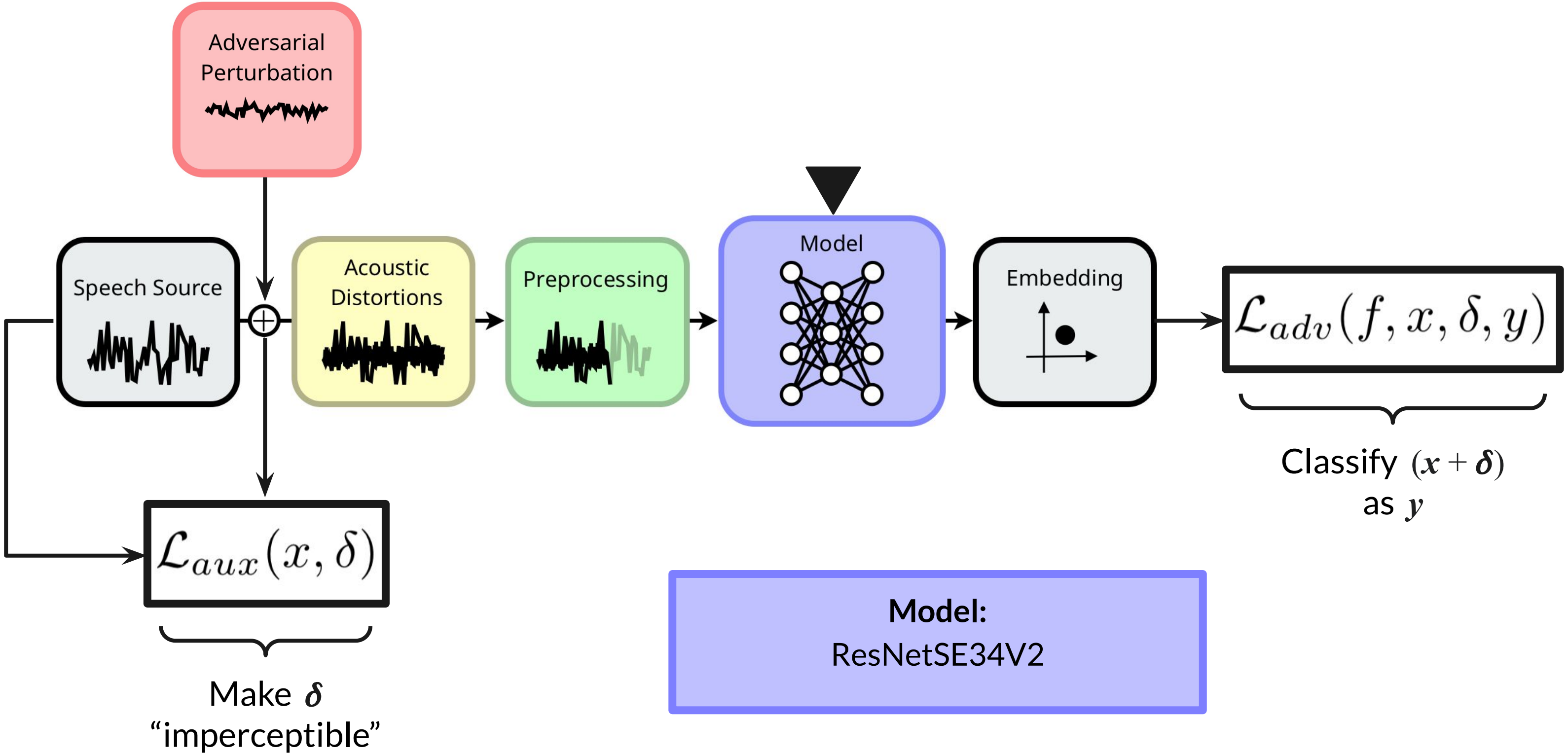


# Let’s Attack a Voice Interface: System Design



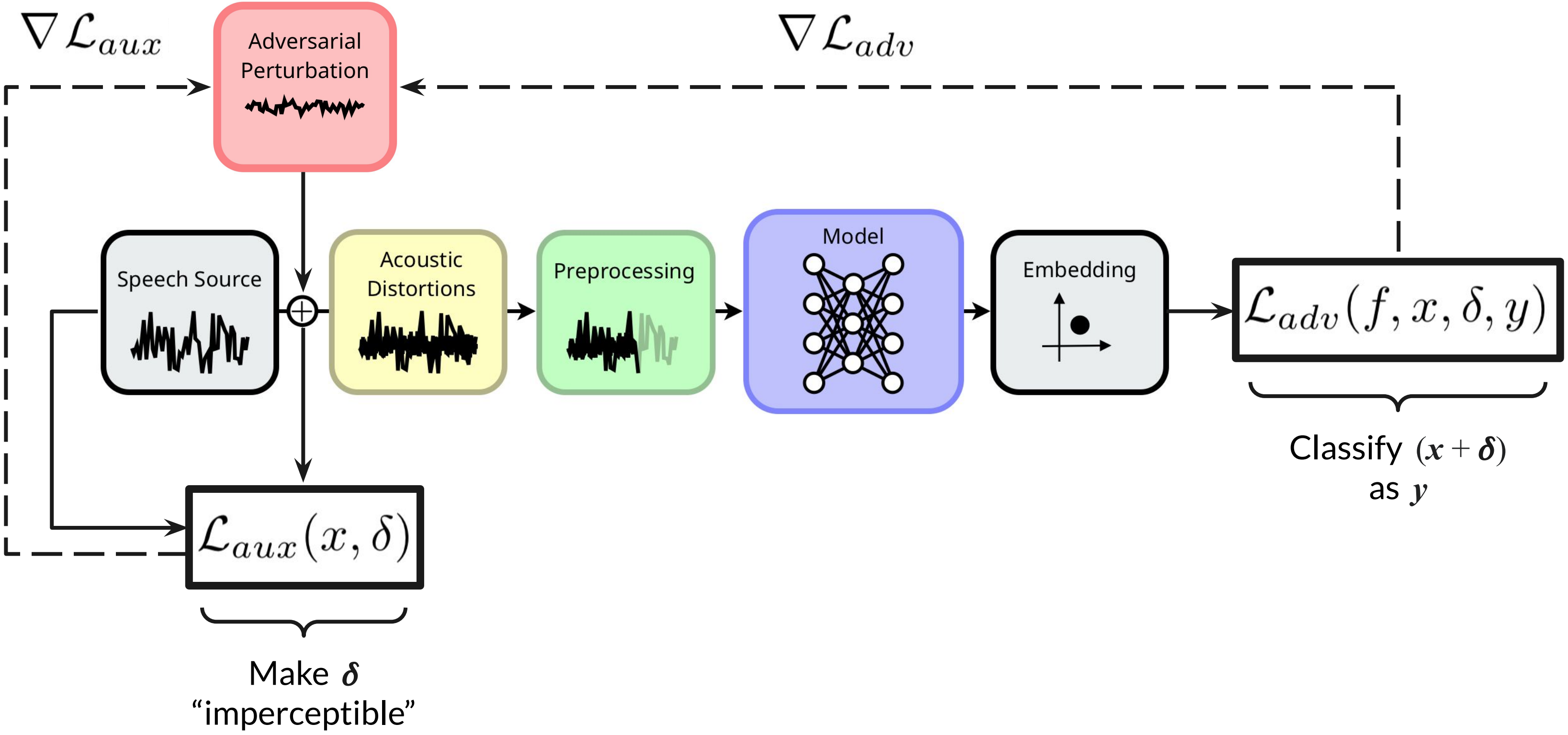

**Preprocessing:**
normalization
voice activity detection (VAD)

# Let’s Attack a Voice Interface: System Design

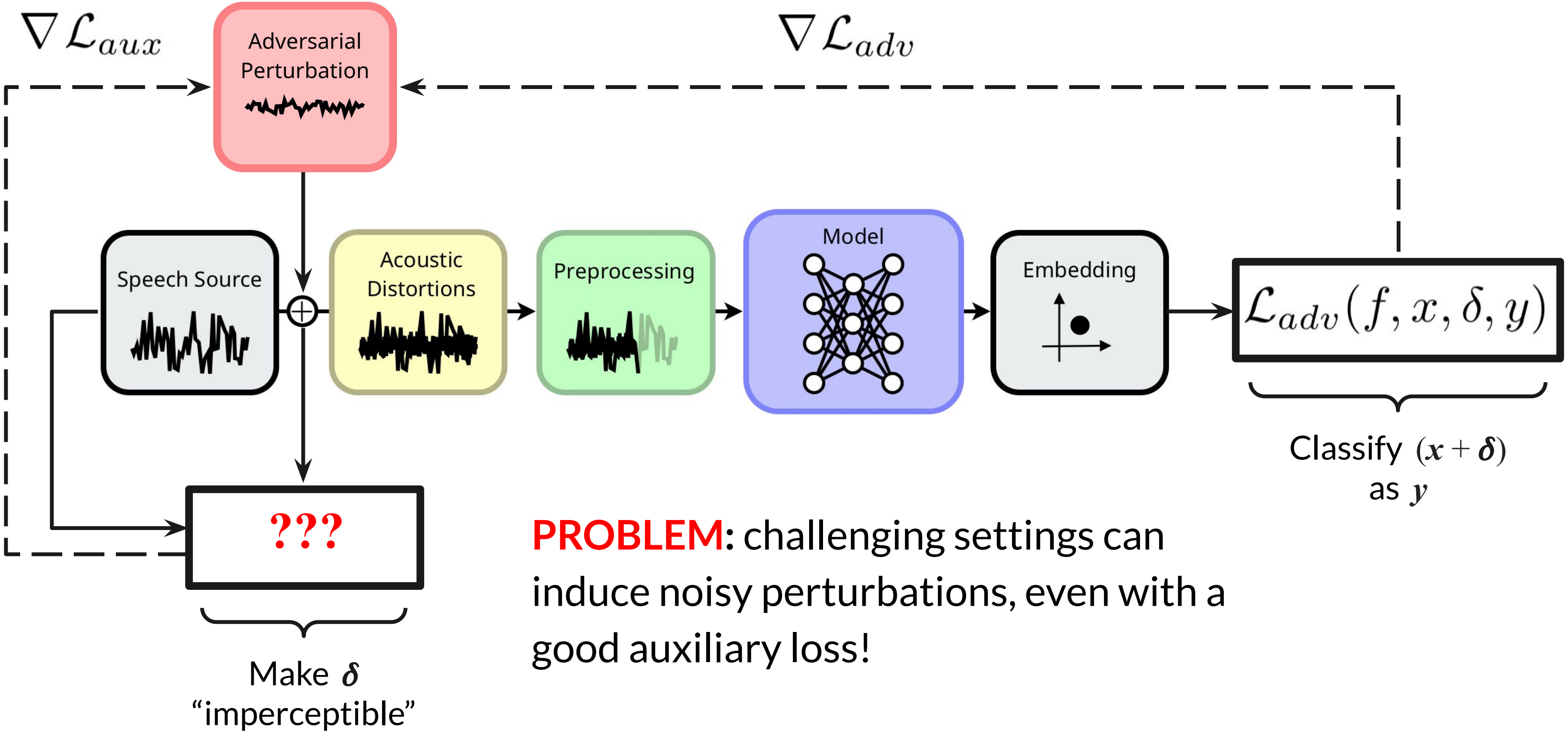



**Model:**
ResNetSE34V2

# Let's Attack a Voice Interface: System Design



# Let's Attack a Voice Interface: The Noise Issue




**PROBLEM**: challenging settings can induce noisy perturbations, even with a good auxiliary loss!

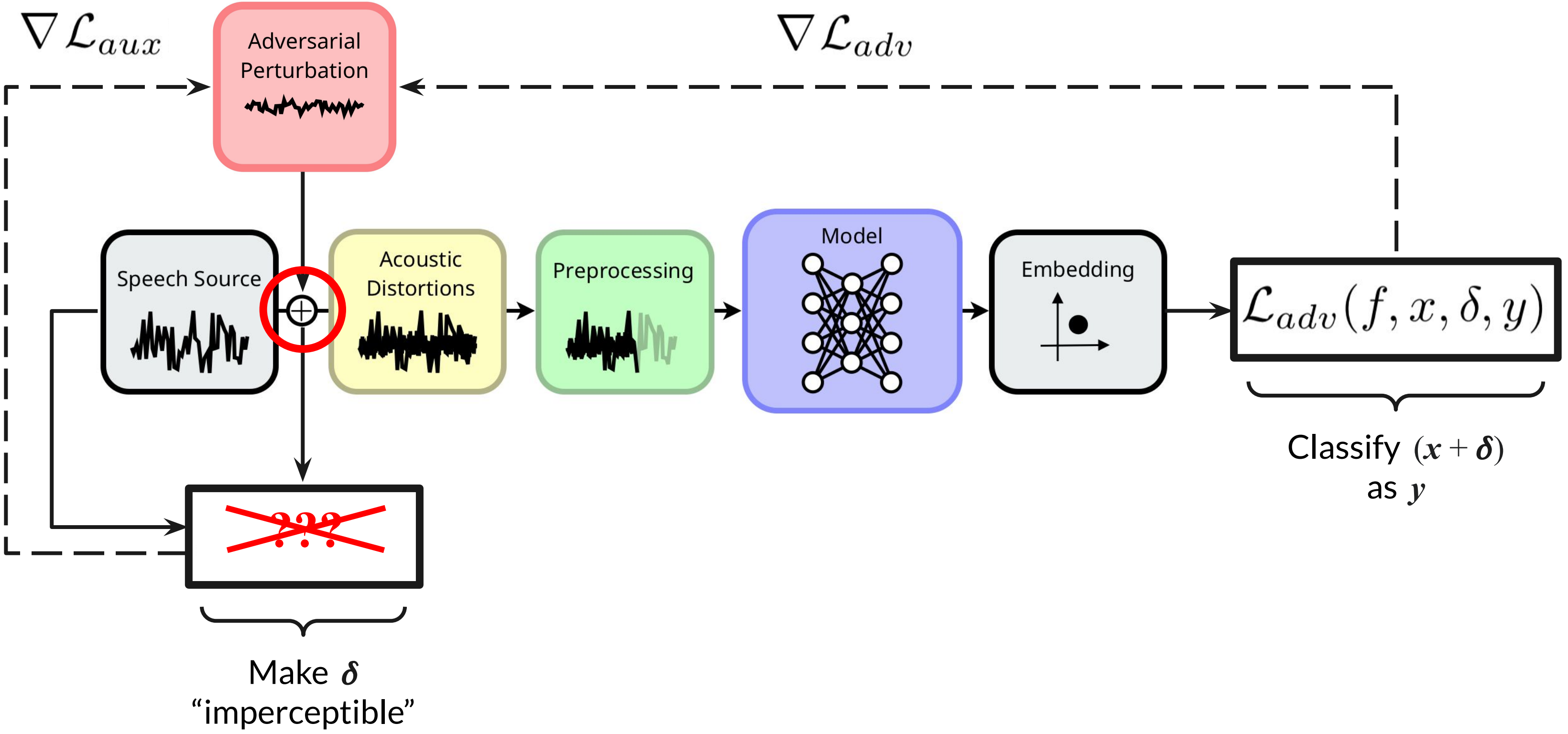
# Let’s Attack a Voice Interface: Pick an Attack

Qin et al. (2019): speech recognition

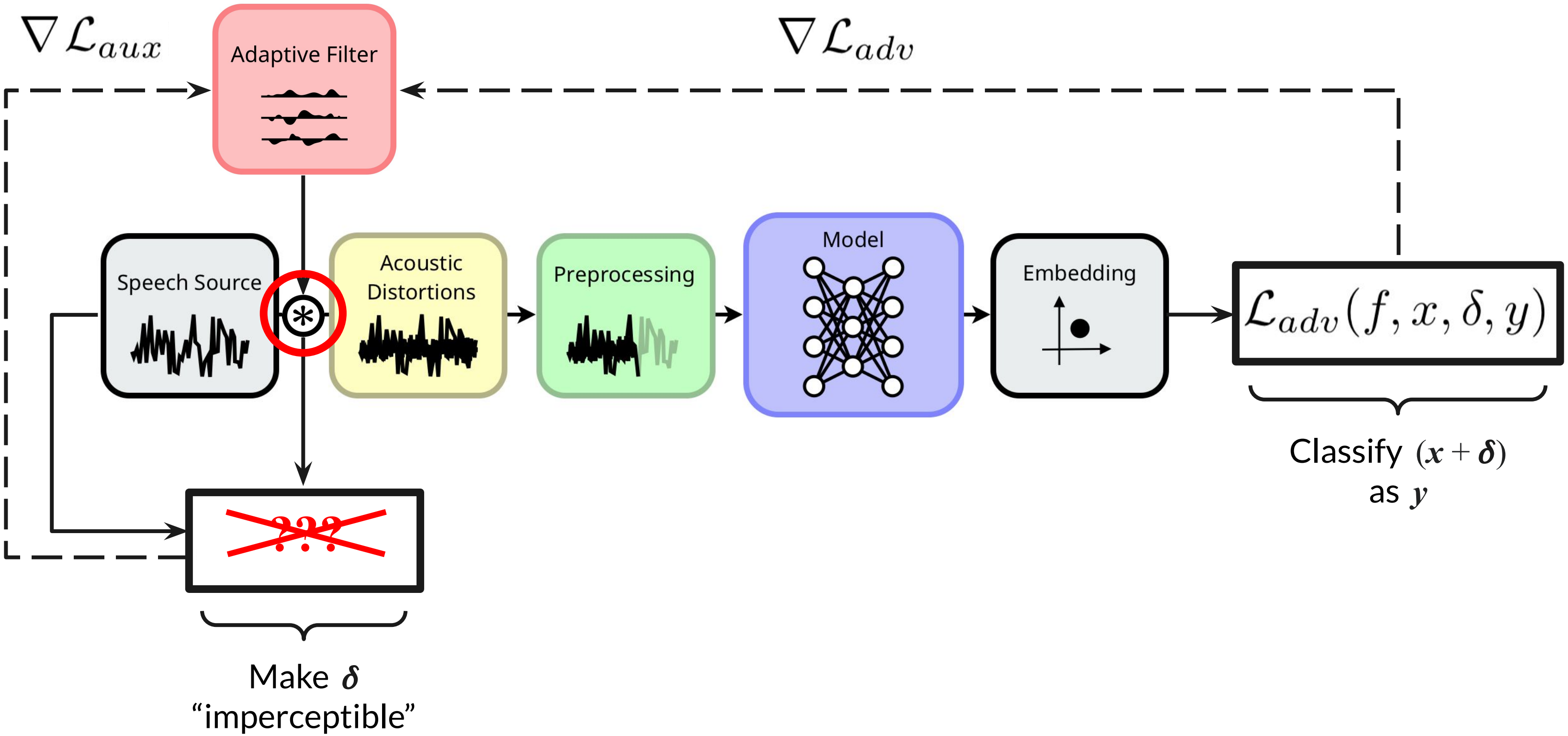
Li et al. (2020): speaker recognition

Chen et al. (2020): speech recognition

# Let's Attack a Voice Interface: System Design



# Let's Attack a Voice Interface: Adaptive Filter Attack

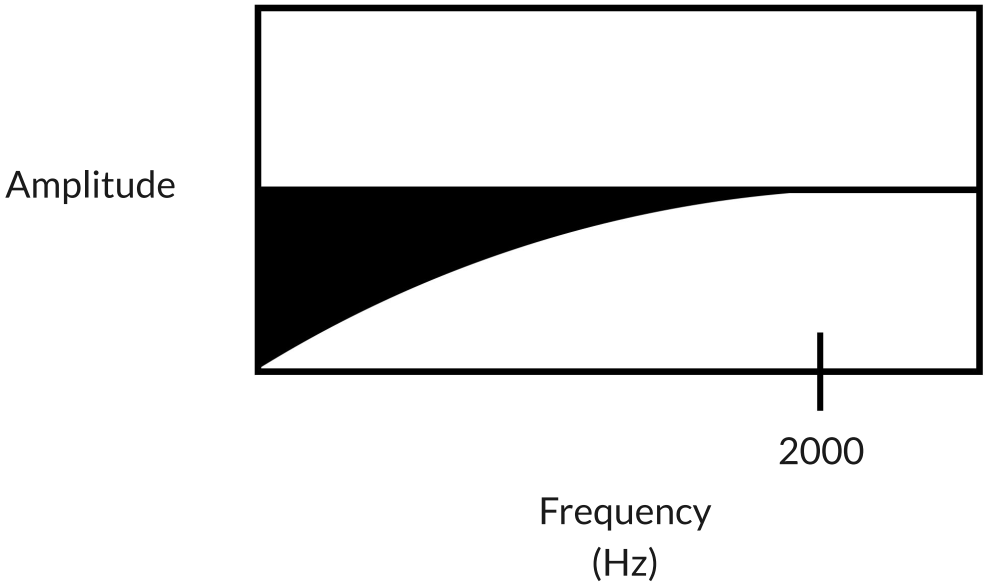


# Adaptive Filters Let Us Shape Frequency Content

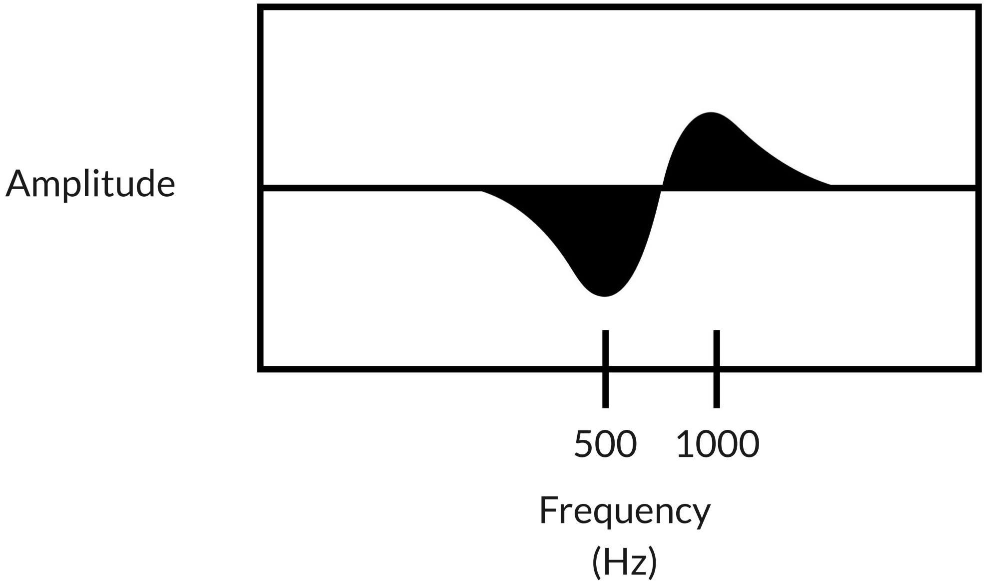
Amplitude

Frequency
(Hz)

# Adaptive Filters Let Us Shape Frequency Content

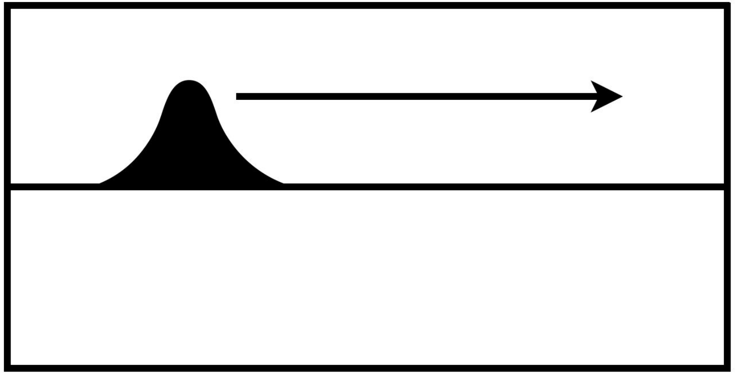


# Adaptive Filters Let Us Shape Frequency Content



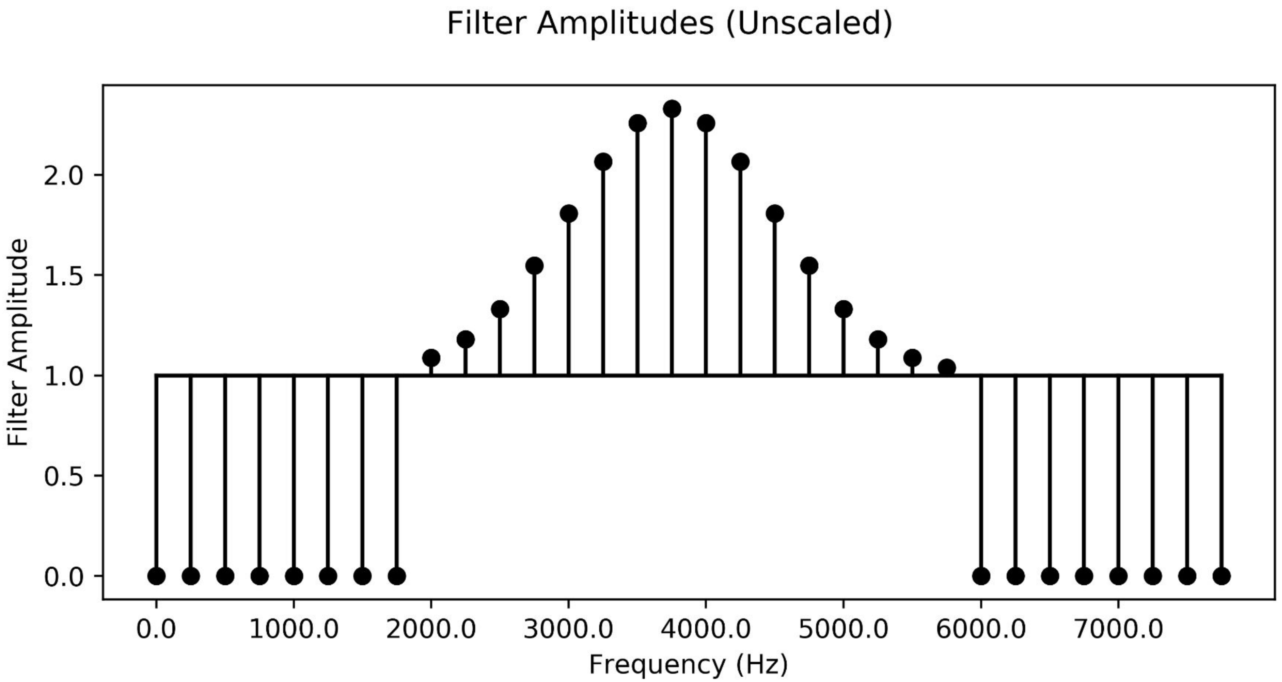
# Adaptive Filters Let Us Shape Frequency Content

Amplitude



Frequency
(Hz)

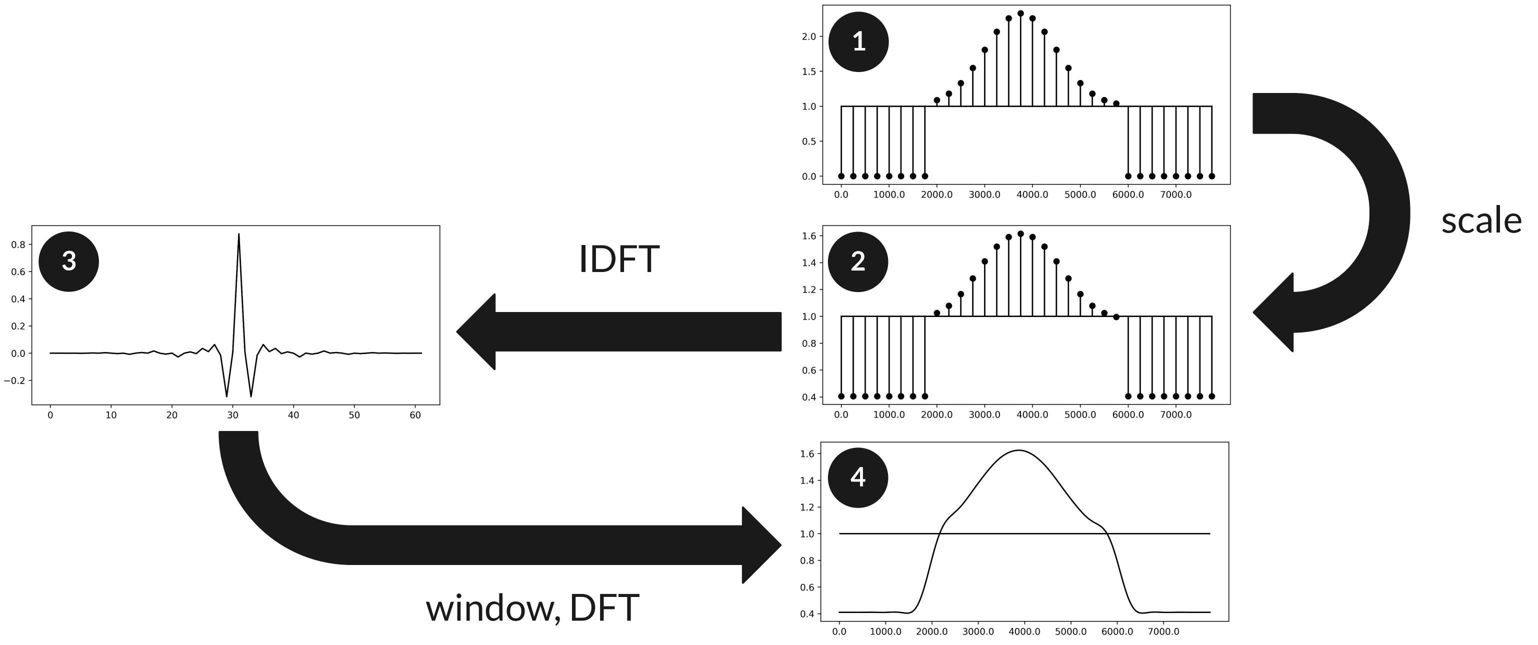
# Adaptive Filters Let Us Shape Frequency Content



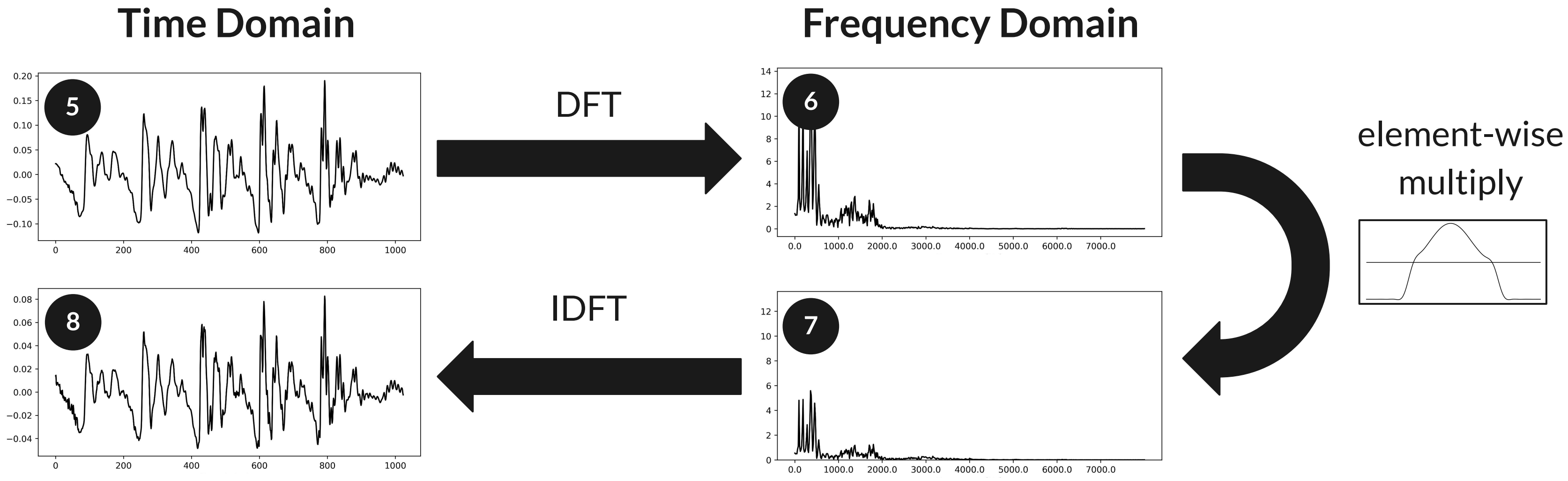
# Adaptive Filters Let Us Shape Frequency Content

## Time Domain

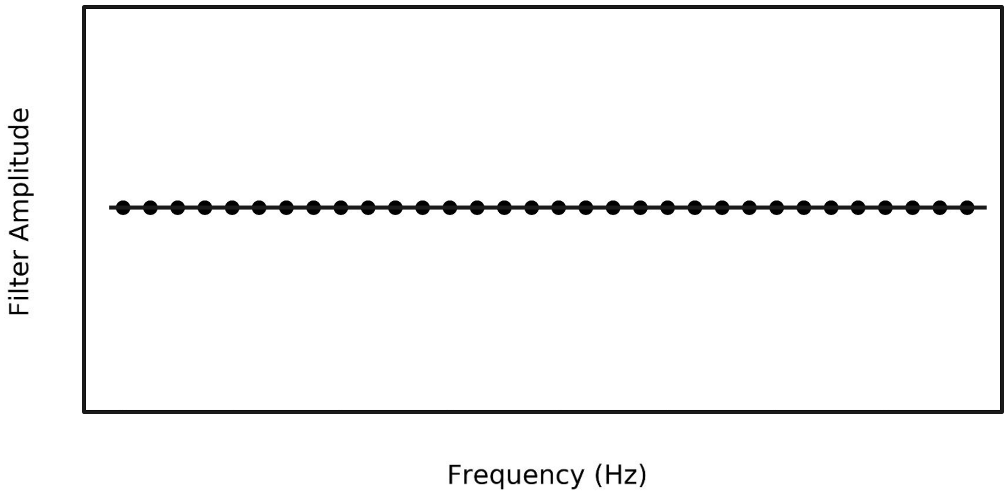
## Frequency Domain



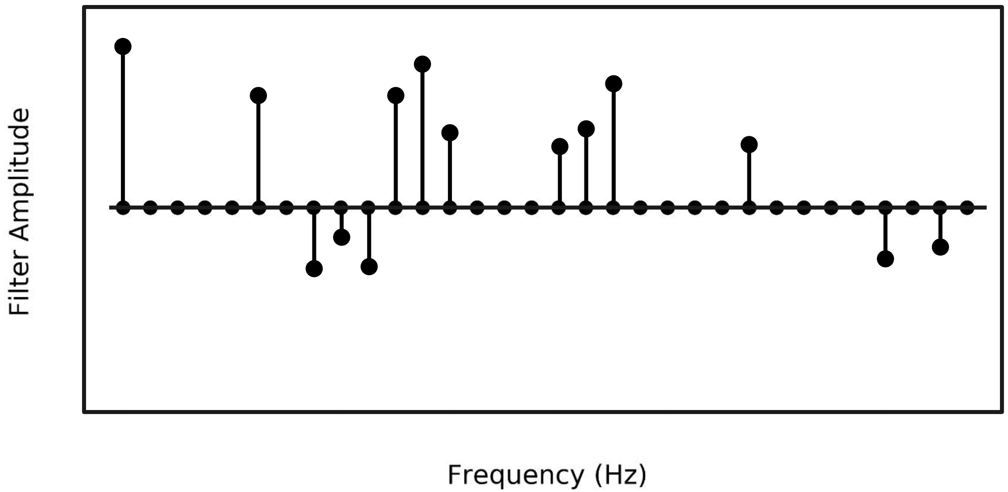
# Adaptive Filters Let Us Shape Frequency Content



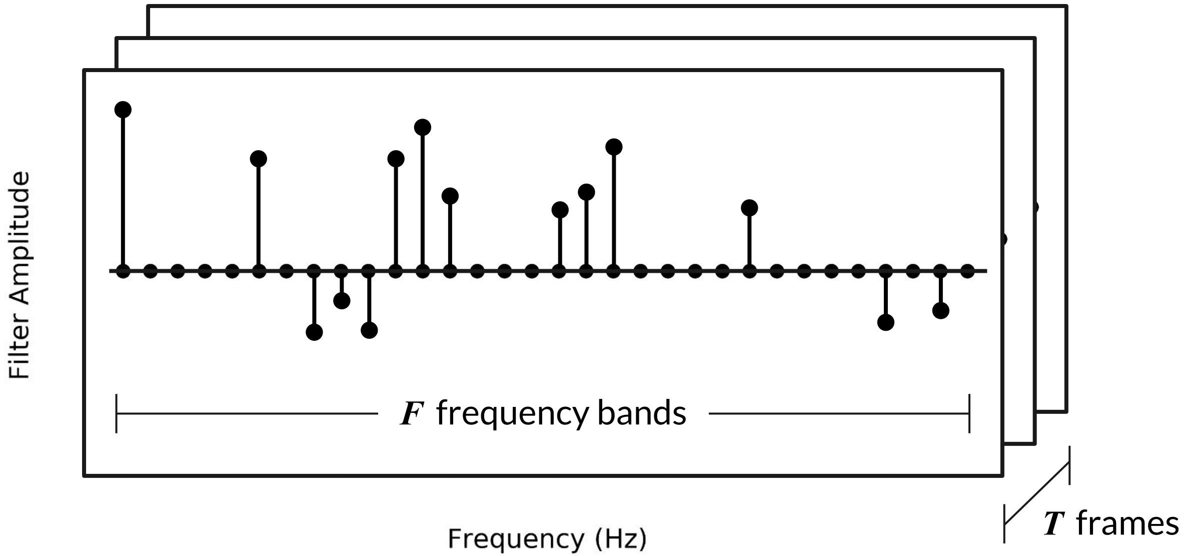
# Adaptive Filters Let Us Shape Frequency Content



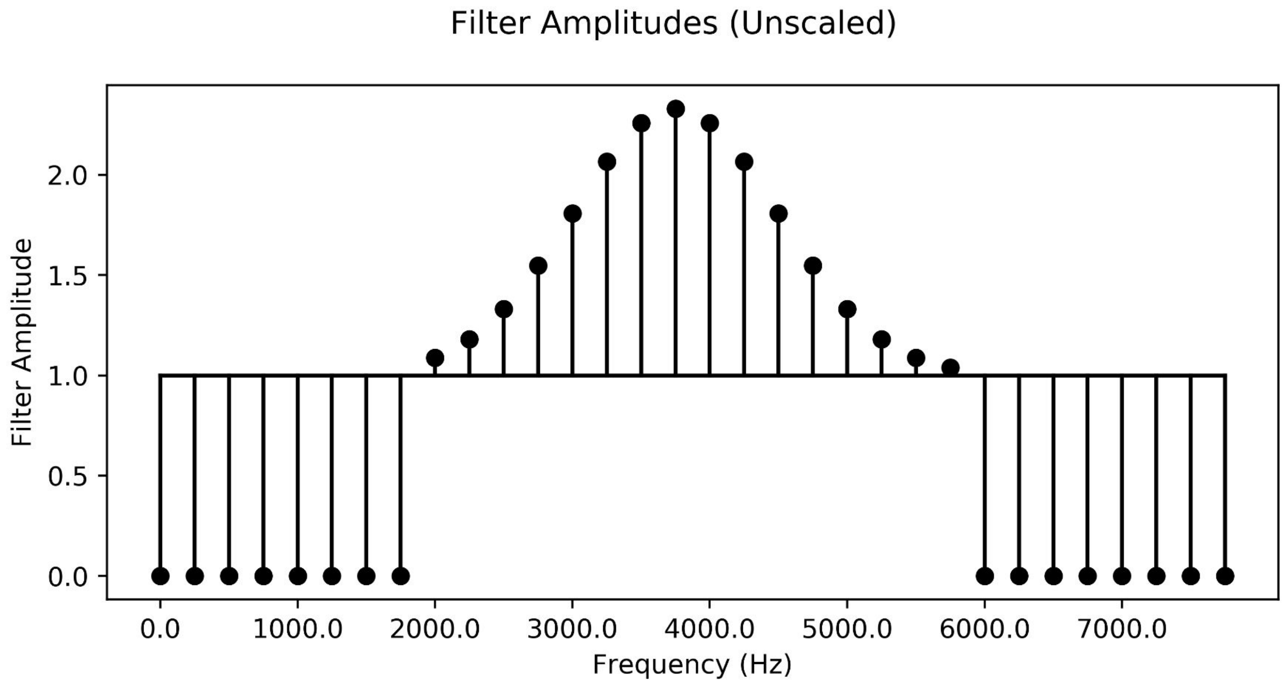
# Adaptive Filters Let Us Shape Frequency Content



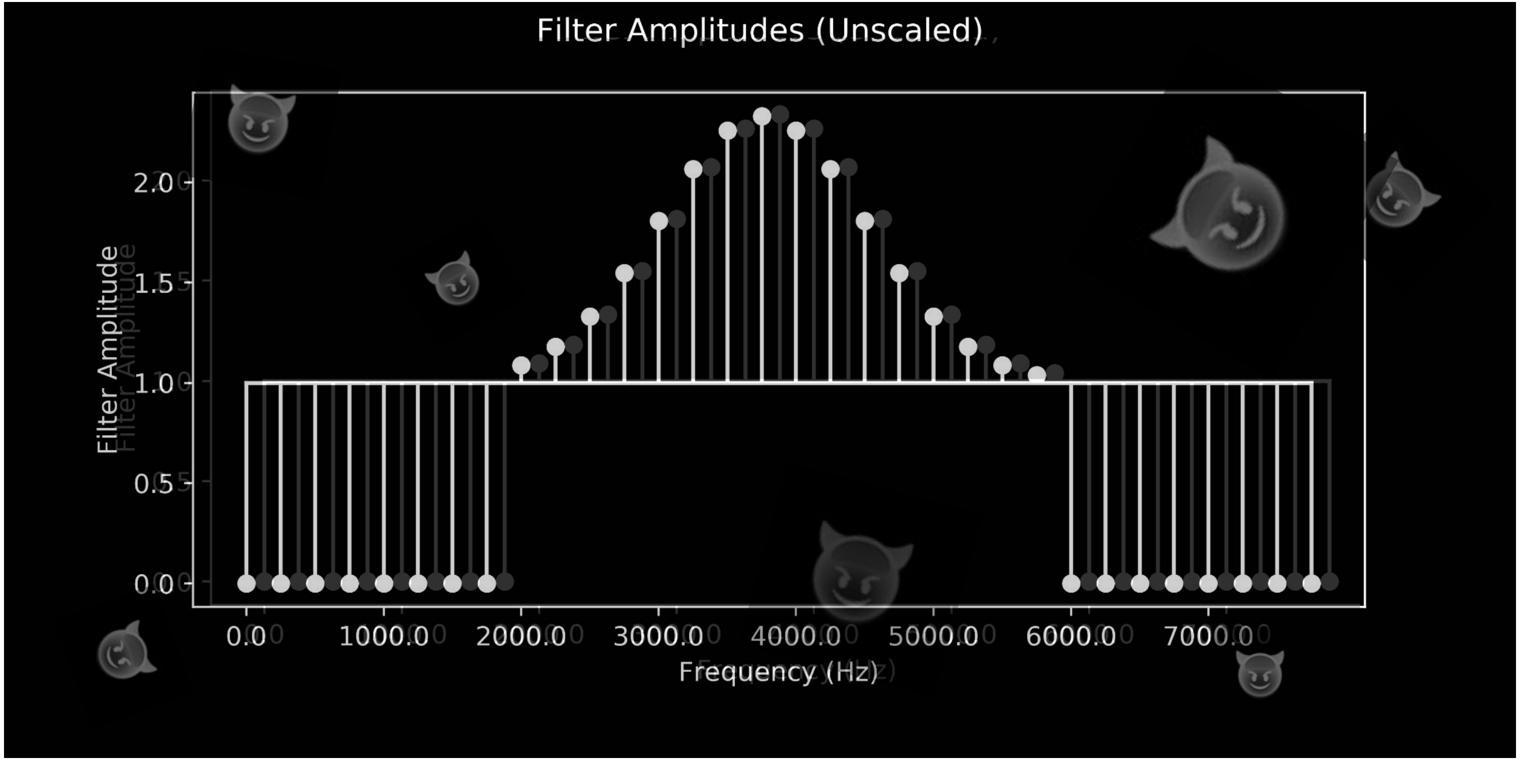
# Adaptive Filters Let Us Shape Frequency Content



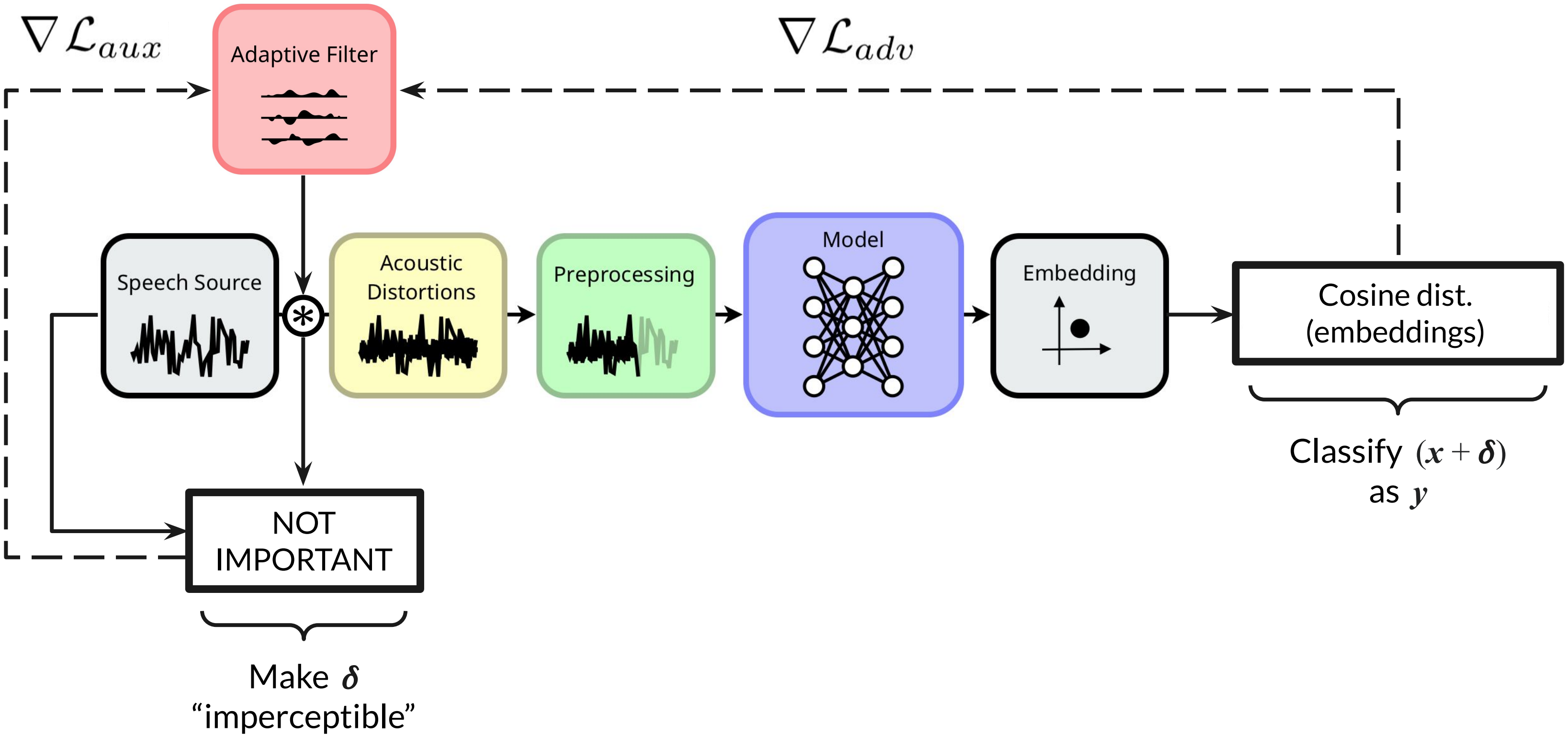
# Adaptive Filters Let Us Shape Frequency Content



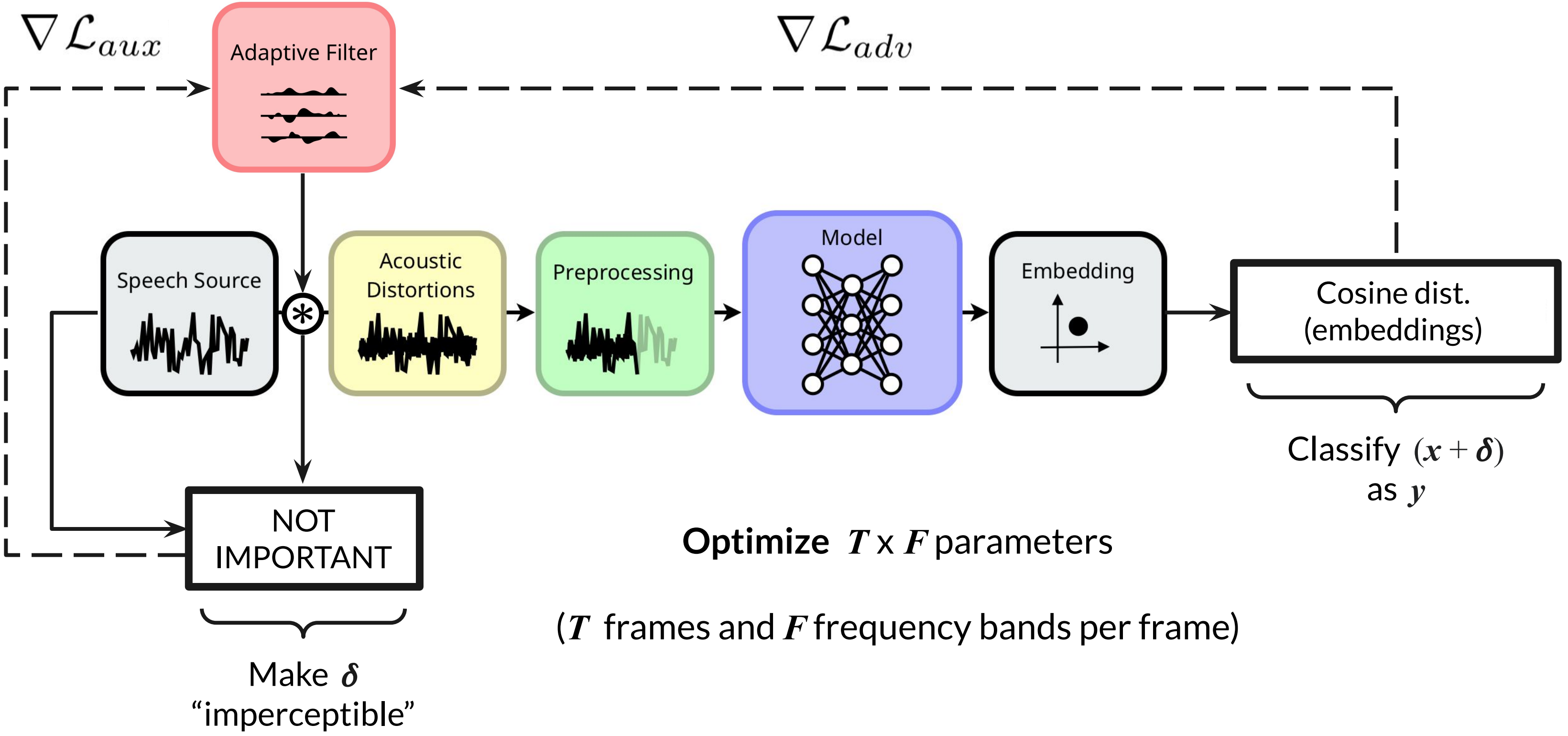
# Adaptive Filters Let Us Shape Frequency Content



# Let's Attack a Voice Interface: Adaptive Filter Attack



# Let’s Attack a Voice Interface: Adaptive Filter Attack

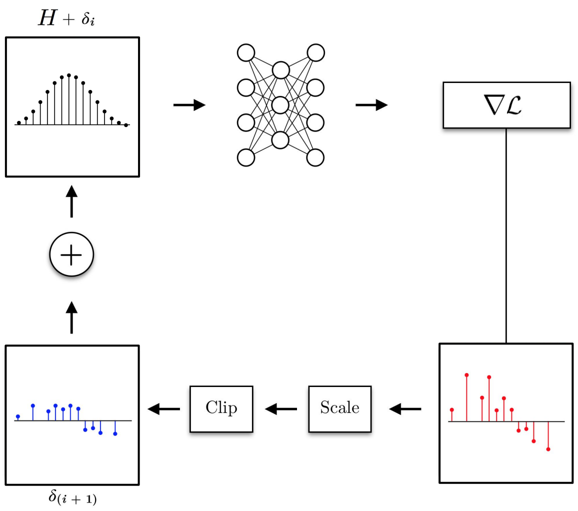



**Optimize** $T$ x $F$ parameters

($T$ frames and $F$ frequency bands per frame)

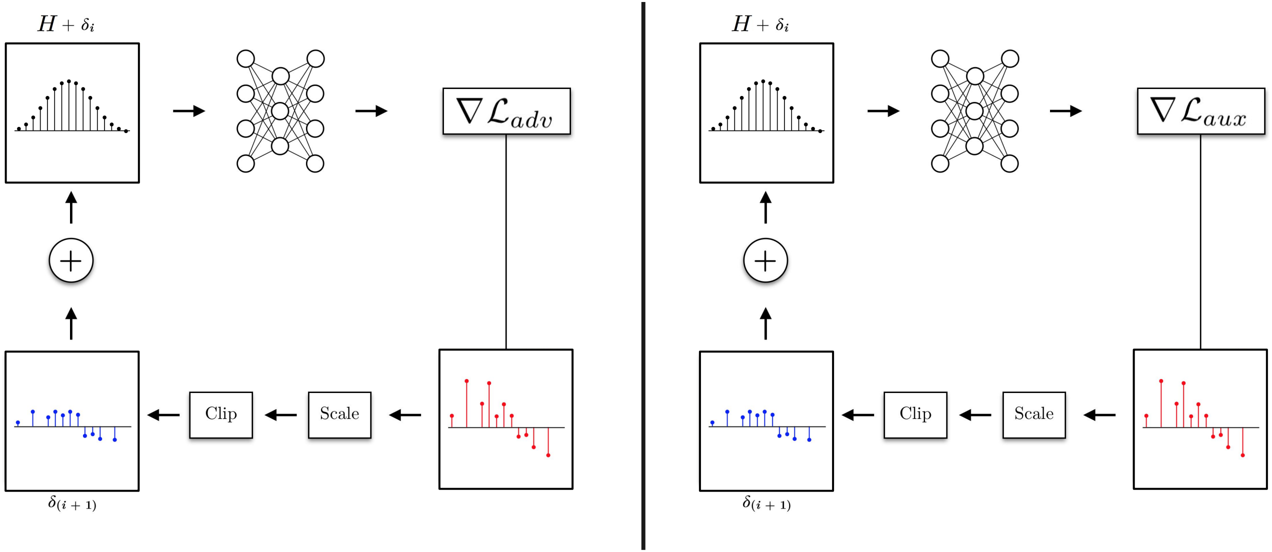
# Let's Attack a Voice Interface: Adaptive Filter Attack

Recall the iterative adversarial optimization procedure we discussed earlier.



# Let's Attack a Voice Interface: Adaptive Filter Attack

*Selective projected gradient descent* (Bryniarski et al. 2021) - break up the updates



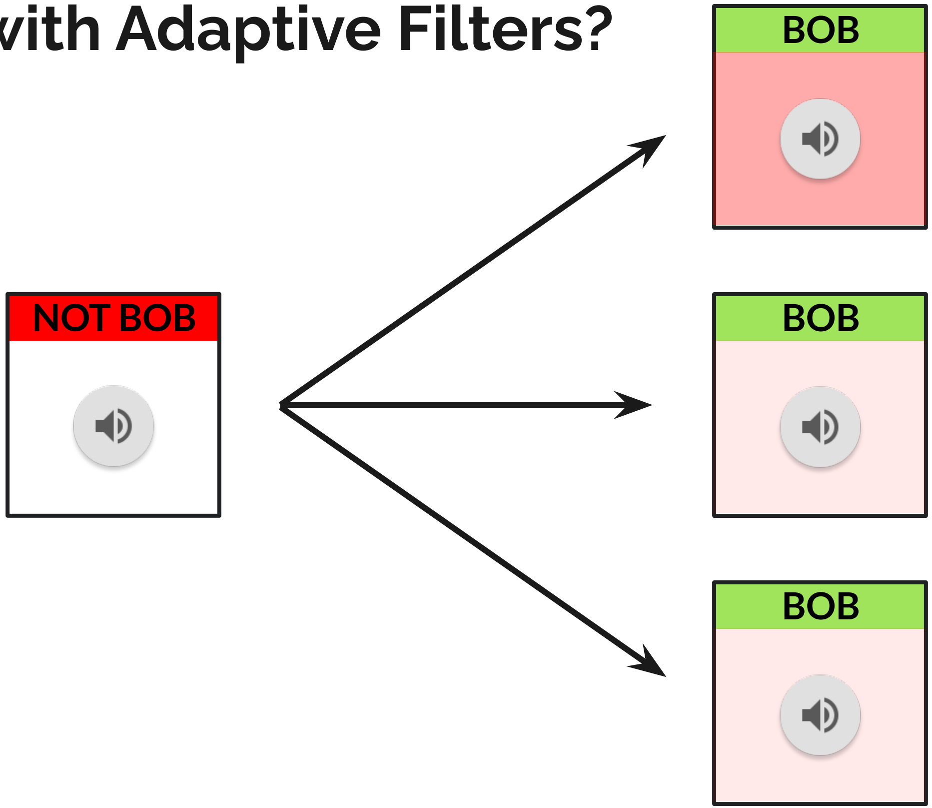

$\mathcal{L}_{adv} > 0$ $\qquad$ $\mathcal{L}_{adv} \leq 0$

# Why Attack with Adaptive Filters?

## Why Attack with Adaptive Filters?

1. Introducing perturbations at the filter representation, rather than the waveform, avoids noise-like artifacts

# Why Attack with Adaptive Filters?




"Generic"

89% effective

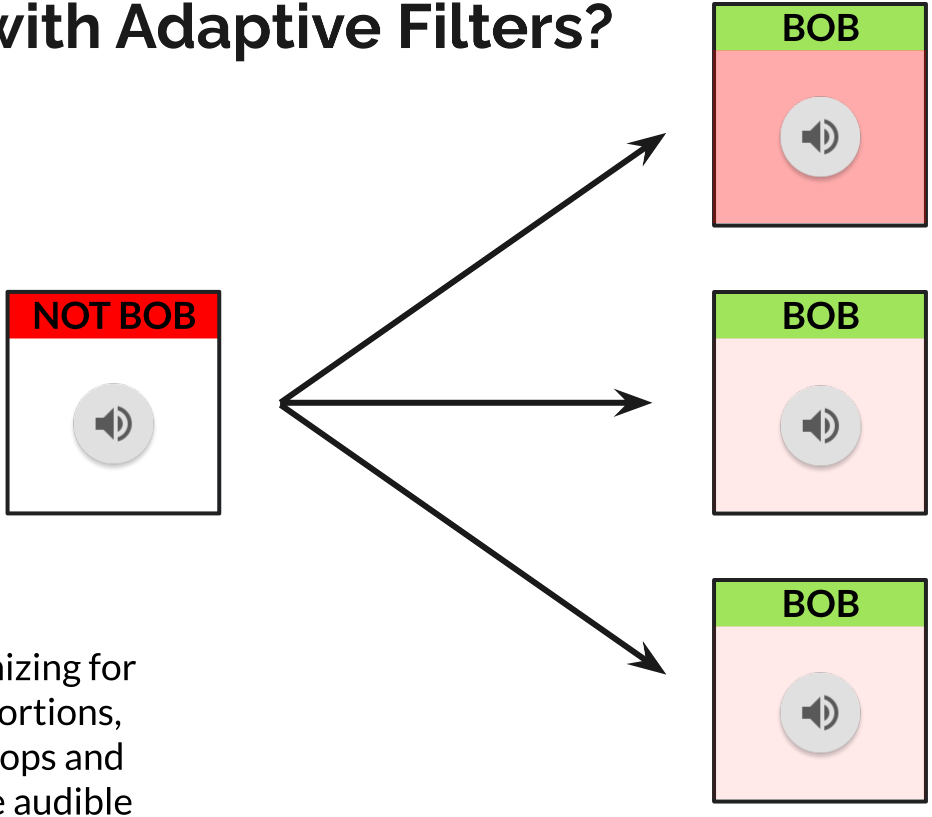
Qin et al.*

93% effective

Adaptive Filtering

95% effective

# Why Attack with Adaptive Filters?




"Generic"

86% effective

Qin et al.*

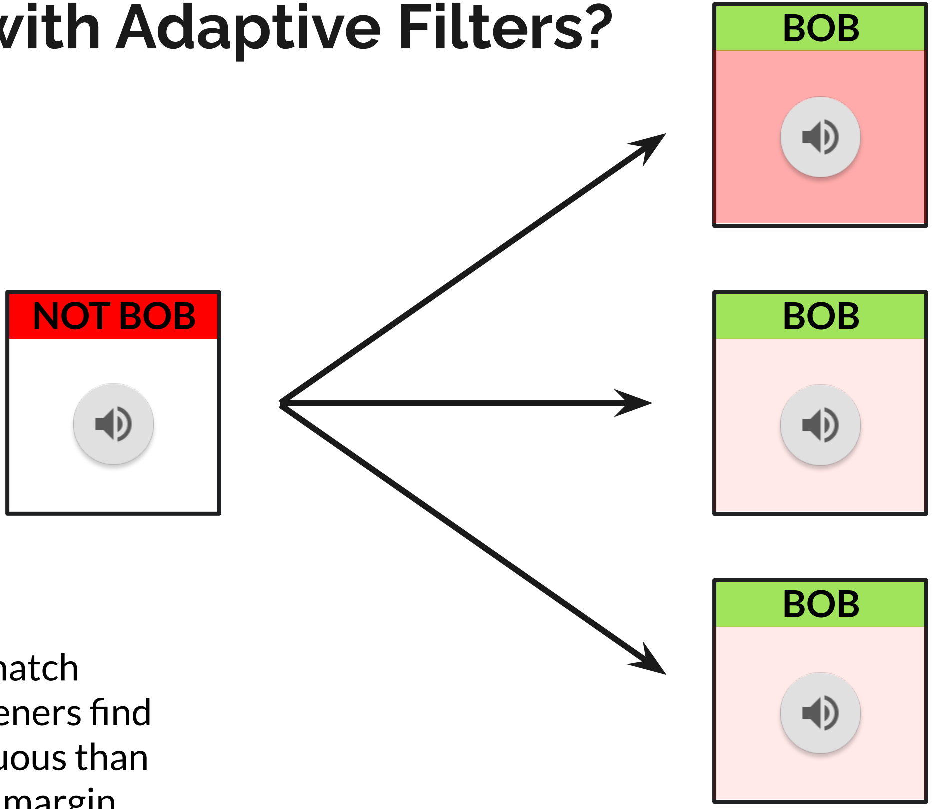
90% effective

Adaptive Filtering

93% effective

In general, when optimizing for more challenging distortions, attack success rate drops and artifacts become more audible

# Why Attack with Adaptive Filters?




**"Generic"**

89% effective

**Qin et al.***

93% effective

**Adaptive Filtering**

95% effective

**User Study:** if we match effectiveness rates, listeners find our attack less conspicuous than Qin et al.* by a 2-to-1 margin

# Why Attack with Adaptive Filters?

| | Waveform $L_\infty$ | Waveform $L_2$ | **Perceptual Study**<br>Forced Choice |
|---|---|---|---|
| Qin et al.* | 0.08 | 1.97 | 34.1% |
| Adaptive Filtering | 0.23 | 6.59 | 65.9% |

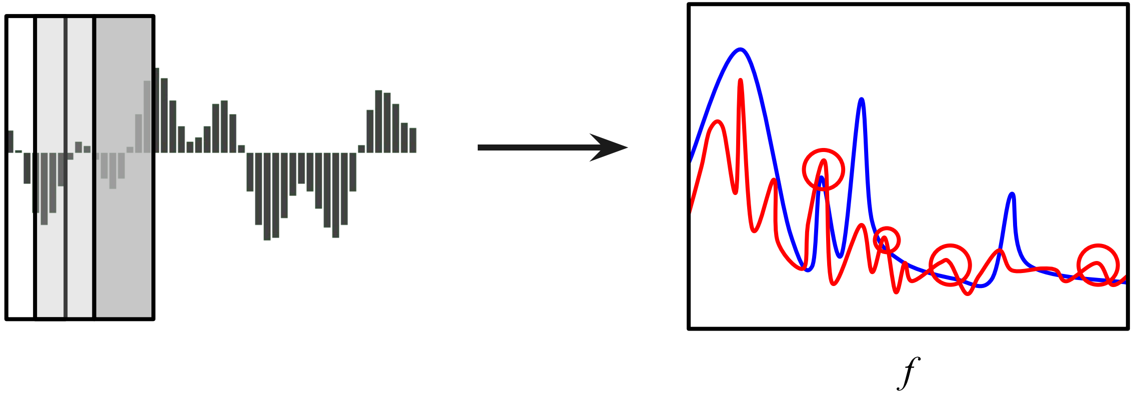
# Why Attack with Adaptive Filters?

| | Waveform $L_\infty$ | Waveform $L_2$ | **Perceptual Study** Forced Choice |
|---|---|---|---|
| Qin et al.* | -- | -- | -- |
| Adaptive Filtering | **2.88x** | **3.35x** | **1.93x** |

## Why Attack with Adaptive Filters?

2. When we use filters, we do not need a complex perceptual loss to produce inconspicuous attacks

# Why Attack with Adaptive Filters?




**Two-stage frequency-masking attack**: Qin et al. (2019), Szurley & Kolter (2019), Dörr et al. (2020), Wang et al. (2020)

# Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. *FoolHD*, *PhaseFool*, *Adversarial Music*)

# Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. *FoolHD*, *PhaseFool*, *Adversarial Music*)

We plan to explore filter-based attacks against more robust speaker verification pipelines, as well as other speech systems

# Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. *FoolHD*, *PhaseFool*, *Adversarial Music*)

We plan to explore filter-based attacks against more robust speaker verification pipelines, as well as other speech systems

We also plan to explore the implications of this work for improving the robustness of audio models against large-magnitude frequency-domain perturbations

# Adversarial Attacks in the Audio Domain with Adaptive Filtering

Patrick O'Reilly[1], Pranjal Awasthi[2], Aravindan Vijayaraghavan[1], Bryan Pardo[1]

https://interactiveaudiolab.github.io/project/audio-adversarial-examples.html

1. Northwestern University
2. Google Research

# Thanks!